The Design and Analysis of Message Authentication and Authenticated Encryption Schemes

Atul LUYKX

Examination committee:
Prof. dr. ir. Pierre Verbaeten, chair
Prof. dr. ir. Bart Preneel, supervisor
Dr. Elena Andreeva
Prof. dr. ir. Luc Van Eycken
Prof. dr. ir. Vincent Rijmen
Prof. dr. ir. Joan Daemen
(ST Microelectronics, Belgium, and University of Nijmegen, the Netherlands)
Dr. Martijn Stam
(University of Bristol, UK)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science: Electrical Engineering

June 2016

Preface

Any type of success I might have had throughout the years is only due to the help of many people, some of whom I thank below.

First I would like to thank the Flanders Innovation and Entrepreneurship agency (IWT) for their financial support which made this thesis possible.

I would also like to thank my adviser, Bart Preneel, for giving me the opportunity to freely perform research in an excellent environment, and for his advice along the way. Then, Vincent Rijmen for the nice conversations we had and for his very useful advice as well. Joan Daemen I would like to thank not only for being on my jury, but also for the discussions throughout the years. I would also like to express my gratitude to Luc Van Eycken and Martijn Stam for providing valuable feedback on my thesis and attending my defences, and Pierre Verbaeten for chairing the jury. Finally, Elena I am very grateful to for being there since the beginning, giving me good feedback, and always looking out for me.

Bart Mennink also guided me from the beginning. He taught me what real work ethic is, and I learned a lot from his expertise. I am grateful to him for the many collaborations and papers we wrote together, for keeping me sharp, and his mentorship.

My PhD started by collaborating with a large group of people, which included Andrey Bogdanov, Nicky Mouha, and Kan Yasuda. I would like to thank Andrey for sharing his insight into finding significant research, and for also inviting me to DTU. Nicky I have always had great and interesting conversations with, which lead to fruitful collaboration. Nicky's advice and guidance helped me a lot throughout my PhD. Then there is Kan, whose amazing insight and guidance pushed me to think beyond what I normally would have. I doubt my PhD would have succeeded without his mentorship.

I would like to thank my co-authors Begül Bilgin, Philipp Jovanovic, Alan

Szepieniec, Elmar Tischhauser, and Laura Winnen for their collaboration and insight. Guy Barwell, Stefan Köbl, Martin Lauridsen, and Tyge Tiessen I would like to thank for the nice discussions we had at the various conferences and summer schools throughout my PhD.

I am very grateful to NTT and Abe-san for providing me with the opportunity to experience working in Japan with the many wonderful people over there. And of course, Ryan, for giving us a great time while living there.

Orr I would like to thank for giving me a great opportunity in Haifa, and for collaborating with me; I felt very welcome in Israel, especially with Muhammad showing me around. I am also very grateful to Tomer and Michal for taking care of me, not just during my stay in Israel, but also throughout my PhD. Tomer has always kept me on my toes, and made sure that I was strong.

Within COSIC I would like to thank first and foremost Péla, Wim, and Elsy who helped me through the non-research aspects of my PhD. Furthermore, I would like to thank my office-mates who provided pleasant environments allowing me to keep my sanity: András, Bing Sun, Nikos, Qingju, Victor, Yoni, Zhiqiang, and Nikos's plant. And I sincerely apologize to all the other people in COSIC whom I did not include here, but made my stay wonderful via the many alma lunches, barbeques, Friday beers, and karting trips.

I am very grateful that Eva joined me halfway through my PhD, making the second half of my PhD fun, and more meaningful. Arun, Uncle Rakesh, and Uncle Ranjan I am grateful to for getting me interested in mathematics and programming, and for guiding me throughout the years, and Aditi for looking out for me and being patient with me. Finally, I am eternally grateful to my parents.

Atul Luykx
Leuven, June 2016

Abstract

Awareness of the significance of securing communication and data has increased dramatically due to the countless examples showing that systems with little or no protection can and will be attacked. Lack of adoption, or improper use of strong cryptographic techniques could be attributed to the fact that cryptographic solutions are not efficient enough, impose impractical constraints on their use, or their analysis does not align with how they are used in practice. This thesis studies message authentication and authenticated encryption algorithms, which are symmetric-key solutions to providing data integrity and confidentiality. A formal study is performed of how security degrades when authenticated encryption algorithms are implemented in environments where theoretical assumptions might not be met, the so-called nonce abuse and release of unverified plaintext settings. Designs for authenticated encryption schemes are analyzed, including our designs COPA and COBRA, while keeping efficiency constraints in mind. Additionally, limits imposed by constrained environments, which commonly appear in applications for the internet of things, are considered, and discussed in the context of message authentication algorithms. A new design is introduced, LightMAC, which enables keys to be used longer than typically possible, and an existing construction, PMAC, is analyzed in depth for its potential to provide more security than what was commonly thought.

Beknopte samenvatting

Het besef van het belang van data- en communicatie-beveiliging is sterk toegenomen vanwege het stijgend aantal aanvallen op systemen met weinig of geen bescherming. Gebrek aan, of foutief gebruik van sterke cryptografische algoritmes kan te wijten zijn aan het feit dat bestaande oplossingen niet efficiënt genoeg zijn, onpraktische beperkingen hebben, of het feit dat hun analyse niet overeenkomt met gebruik in de praktijk. Deze thesis bestudeert de symmetrische-sleutel algoritmes om integriteit en vertrouwelijkheid van data te verzekeren, namelijk, bericht-authenticatie en geauthenticeerde encryptie-schemas. De manier waarop de beveiliging van geauthenticeerde encryptie-schema's degradeert wanneer ze geïmplementeerd worden in omgevingen waar theoretische veronderstellingen niet noodzakelijk gerespecteerd worden, de zogenaamde *nonce*-misbruik en *release of unverified plaintext* omgevingen, wordt formeel bestudeerd. Ontwerpen van geauthenticeerde encryptie-algoritmes worden geanalyseerd met oog op efficiëntie. Verder worden beperkingen van bericht-authenticatie algoritmes in omgevingen met implementatie-beperkingen, zoals gevonden in toepassingen van de *internet of things*, besproken. Een nieuw ontwerp dat sleutels langer kan gebruiken dan standaard algoritmes, LightMAC, wordt geïntroduceerd, en een uitgebreide veiligheidsanalyse toont aan dat een bestaande constructie, PMAC, meer beveiliging zou kunnen aanbieden dan oorspronkelijk gedacht.

Contents

List of Figures

Chapter 1

Introduction

1.1 Communication Challenges

Nineteenth-century Flemings faced a world going through significant changes. A recent revolution had created the country of Belgium in which they now lived, a potato disease running through Europe was destroying crops resulting in thousands of deaths, and the industrial revolution was forcing people to re-evaluate how labour was done. On the list of major concerns for the typical Fleming, privacy would not have ranked high. The speed and scope of communication simply would not have exposed privacy threats far beyond his or her immediate surroundings, since the vast majority of communication would have been face-to-face, and the most advanced technology, the telegraph, would have seen little use by the Fleming.

Twenty-first-century Flemings might not have developed intuition beyond the nineteenth century concerning privacy, and assume that information travels only to the intended recipient, with little leakage otherwise. However, this intuition could not be further from the truth. Basic mobile-phone usage broadcasts all communication wirelessly over a large range, allowing people with an antenna to intercept, and even impersonate providers. Sending emails is more akin to sending postcards written in pencil: anyone can read the contents, and modify the text without detection. Connecting to bank websites could pose significant threats, with impersonation a real possibility.

Furthermore, these methods of communication only scratch the surface of information that could be compromised. The increasing prevalence of devices connected to the internet, more commonly known as the *internet of things*,

further exposes a wealth of information to interested parties, by connecting home printers, medical devices, and even baby monitors to the internet.

Although some people might argue they have nothing to hide, most people, when given the option, would rather not have exposure similar to a reality-TV show.

1.2 Connecting to Facebook

Consider, for instance, a user connecting to the online social network Facebook via a browser. As recently as 2010 the connection would have been mostly performed using the *Hypertext Transfer Protocol*, or HTTP, a method of retrieving websites from a server. After requesting the Facebook login page, the user would type in her information, which would be formatted appropriately so that the server can interpret the login request. The data would be passed to the user's internet service provider, and subsequently a path of nodes would be found through the internet, enabling delivery of the data to its destination. Upon receipt of the correct login information, the server sends back the home page of the user's Facebook account.

HTTP is simply a language in which the user's browser and Facebook's server communicate, and therefore its goal is to be as unambiguous as possible. In particular, it makes no guarantees of whether the information received by the server actually comes from the user, nor does it make any claims of whether the information was exposed to all the intermediate nodes over the internet. In fact, in 2010, Tunisian internet service providers exploited these properties to inject malicious code which captured users' Facebook login information [17]. This was during the height of the Tunisian revolution, a period in which Facebook, and social media in general, were being used by protesters to spread uncensored news and organize themselves. Facebook received anecdotal reports of accounts being compromised, but the attacks were otherwise undetected.

1.3 Transport Layer Security

Once Facebook determined the cause of the attacks, they pushed the use of HTTP**S**, which wraps *Transport Layer Security* (TLS) around HTTP. TLS is a protocol which attempts to provide *confidentiality*, or the inability of adversaries to determine the contents of the communication, and *authenticity*, or the inability of adversaries to impersonate, modify, or inject new data during communication. To achieve these goals, TLS uses tools developed within *cryptography*, the study

of efficient methods to ensure that processing and communication of information is only done by authorized entities.

TLS breaks communication down into two parts: the handshake protocol and the record layer protocol. The handshake protocol uses *asymmetric* cryptography to establish initial contact between two communicating parties. Two parties, A and B, that wish to communicate using asymmetric cryptography, each establish public keys, which can be released publicly, and private keys, which are kept hidden. When A wants to send B a message, A looks up B's public key, which it uses to encrypt the message, and sends the result to B. When B receives the encrypted data, it is able to recover the original message using its private key. If the scheme is secure, then no-one besides B will be able to decrypt what B receives.

The strength of asymmetric cryptography is that it allows two parties to communicate securely using only public knowledge, which means it can be used to initiate communication. In fact, TLS really only needs asymmetric cryptography to enable secure communication between users and Facebook. But it is costly to communicate with asymmetric cryptography, which is why the handshake protocol only establishes a shared secret among A and B. This shared secret is then used by the record layer protocol to perform the bulk of the communication. For this, *symmetric* cryptography is used, which provides security assuming that the communicating parties have a shared secret. Symmetric cryptography is not able to establish initial contact, but it is significantly more efficient than asymmetric cryptography.

1.4 Breaking TLS

The use of TLS seemed to stop the Tunisian internet service providers, but more determined adversaries might have used one of the many vulnerabilities present in TLS; see Table 1.1. Guaranteeing the security of the entire TLS protocol is difficult. Various points of failure have been taken advantage of in the attacks against TLS, which could occur in the implementation, the specification or standard, as protocol flaws, or even as cryptographic design flaws. All of these levels need to be secure to guarantee that a particular implementation of TLS is secure.

Often the underlying cryptography is assumed to be the last point of failure, however occasionally cryptographic schemes have been attacked, and when they are compromised, the results can have detrimental effects. An example is the *Flame* malware: undetected for up to five years, it infected private individuals, government organizations, and educational institutions [85]. Flame

Table 1.1: Some attacks against TLS.

Year	Name	Reference
2002	Padding oracle attack	[170]
2009	Renegotiation attack	[148]
2012	Alert attack	[1]
2013	Lucky13	[8]
2014	Triple handshake attack	[41]
2015	Logjam	[4]
2016	SLOTH (Transcript collision attack)	[42]
	DROWN	[18]

uses a weakness in the cryptographic algorithm MD5 to forge a Microsoft certificate. Attacks on MD5 had been studied extensively by the cryptographic community [46, 173], but Flame used a new attack [2].

Another example is the insecurity of the IEEE 802.11 WEP protocol, used for wireless networks. When WEP was developed, there were cryptographic schemes providing confidentiality and authenticity separately, but none addressed the issue of combining the two. As a result, WEP provided its own solution. In 2001 it was shown that WEP attained neither confidentiality nor authenticity [16, 55], and the protocol was exploited in 2007 to steal personal data from over 450 000 customers from a retail store [19].

Although both MD5 and the algorithm underlying WEP were known to be weak in the literature, their continued use in practice highlights the importance of designing efficient algorithms which address the needs of users.

1.5 Goals

Motivated by the lack of adoption of strong cryptographic algorithms in practice, we seek new designs, formalizations, and analysis that not only push the limits of efficiency and longevity of cryptographic schemes, but also add robustness so that security is maintained as much as possible in environments which might not have been accounted for in theory. Our research focuses on the design and analysis of symmetric cryptographic algorithms, more specifically, message authentication schemes, which seek to provide data authenticity, also called integrity, and authenticated-encryption (AE) schemes, which aim for both confidentiality and integrity.

Both message authentication and AE schemes form the backbone of security for many different environments. In settings where confidentiality is not necessary, message authentication algorithms provide the most efficient method of ensuring data integrity. Robust message authentication algorithms already exist, making them suitable for many different environments, however efficiency constraints, especially in constrained environments, limit their usability. Our goal is to investigate what the fundamental limits are of how efficient message authentication algorithms can be, and whether new designs can improve upon the state-of-the-art.

AE schemes provide integrity as well, and are necessary anytime confidentiality is needed. However, designing efficient and robust AE schemes is not as straightforward as with message authentication schemes. It is not obvious what type of security definitions are necessary to analyze AE schemes in environments where basic assumptions are broken. Here our goal is to meaningfully model extended security settings in which AE schemes can be tested and proven secure. Then, we aim to explore the efficiency and design constraints that these models impose in order to create algorithms which are robustly secure, while being as efficient as possible.

1.6 Contributions

With respect to message authentication schemes we focus on one class of easily parallelizable algorithms. We introduce the scheme LightMAC, which is a simple and efficient message authentication algorithm able to process data significantly longer than what is typically possible. Then we analyze PMAC, an existing competitor to LightMAC, in order to understand how its longevity compares. This involves finding attacks against PMAC to illustrate its security limits. However, in our exploration we find that determining PMAC's limits is a non-trivial theoretical problem, which we are able to formalize. Nevertheless, we show that one version of PMAC does have an attack, meaning this version's security limits are significantly lower than LightMAC's.

With respect to AE schemes, we investigate known extensions of the basic security model, nonce misuse resistance. We point out that existing definitions of nonce misuse resistance do not align with intuition, and give a new definition which more accurately models what one would expect to happen to security. Furthermore, we introduce a new setting, called the release of unverified plaintext (RUP) setting, which models scenarios that till now have not been accounted for in the literature. Schemes which are designed to be RUP-secure could be used in many applications to increase the robustness of the system. Finally, various

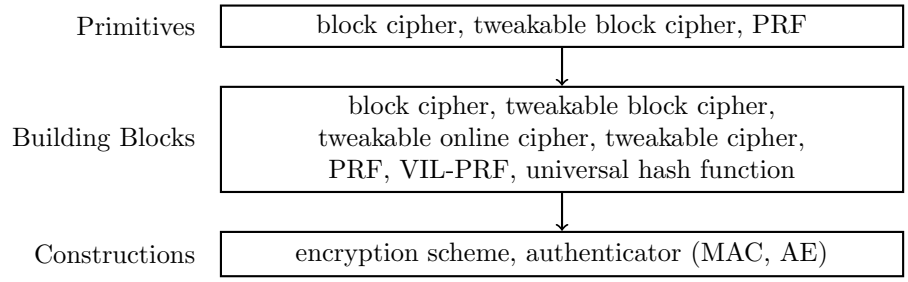


Figure 1.1: Each group of algorithms serves as the tools with which the next group is constructed.

efficient designs are discussed which provide more robustness than conventional AE schemes, including three new designs, COPE, COPA, and COBRA.

1.7 Outline

Chapter 3 reviews basic definitions for confidentiality and integrity. Conventional integrity definitions deal with message authentication schemes and AE schemes separately, but we combine these definitions into one. Chapter 4 then discusses an extension of the basic definitions to describe what happens when a basic assumption about the schemes is no longer met in practice, that is, the so-called nonce misuse setting. The necessary formalization is introduced to describe the setting, followed by a more natural nonce misuse security definition than what is present in the literature. Finally an overview is provided of the connection between the conventional and misuse definitions.

In Chapter 5, tools are presented with which algorithms will be constructed achieving the security definitions of Chapters 3 and 4. All algorithms discussed in the thesis can be categorized as either a primitive, a building block, or a construction, as illustrated in Figure 1.1. Primitive design is a complicated matter and remains out of the scope of this thesis, but years of experience in the cryptographic community has given confidence in the security of primitives such as the Advanced Encryption Standard [67]. These primitives can either be used as building blocks themselves, or to construct more advanced building blocks. Included in the advanced building blocks are the algorithms COPE and COBRA, from our publications published in Asiacrypt 2013 [13] and FSE 2013 [14], respectively, which, at their time of publication, presented the state of the art in efficient algorithms with some resistance to nonce misuse.

The building blocks from Chapter 5 in turn enable us to build constructions which achieve data confidentiality and integrity, as discussed in Chapter 6. Contributions include drawing connections between the different design decisions made for various constructions, and a new application of ciphertext stealing to COPE in order to deal with ciphertext expansion. Furthermore, the algorithm COPA from Asiacrypt 2013 [13] is introduced.

Besides nonce misuse, part of our research also discusses other failures that could happen in practice when implementing authenticated encryption schemes, called the releasing unverified plaintext setting [12]. Chapter 7 places the results from our paper [12] in the context of the framework introduced by Barwell et al. [21], in order to gain insight into the setting.

The thesis is concluded with Chapter 8, which discusses our work on message authentication, and how the improvement of security bounds can have practical impact. This chapter consists mainly of text from our publications on LightMAC [121, 122] and security bounds for PMAC [119, 120], where the works are presented nearly in their entirety with little modification. The emphasis of this chapter is on longevity of schemes as opposed to efficiency or added robustness.

Chapter 2

Preliminaries

In this chapter we describe the basic mathematical definitions necessary for the thesis, and outline some of the most important concepts in order to understand our approach. After covering notation and binary fields, we describe the elements necessary for our security definitions, namely algorithms, adversaries, and success measures. Then we place our approach to security in context by describing reductions, and then efficiency measures. The chapter is concluded with some technical definitions necessary for the proofs.

2.1 Notation

For a set X, X^n is the set of n-length sequences of elements of X, $\mathsf{X}^{\leq n}$ is the set of sequences of length not greater than n, X^+ is the set of finite-length sequences of length at least one, and X^* is X^+ along with the "empty" sequence, usually denoted ε. If $X \in \mathsf{X}^*$, then $|X|$ denotes its length. For $X \in \mathsf{X}$ and $Y \in \mathsf{Y}$, $X \| Y$ and XY interchangeably denote the element $(X, Y) \in \mathsf{X} \times \mathsf{Y}$. Given an element $X = (X_1, X_2, \ldots, X_n) \in \mathsf{X}^n$ and an integer $t \leq n$, then $\lfloor X \rfloor_t$ denotes the first t components of X, that is, $(X_1, X_2, \ldots, X_t)$.

The set of arbitrary length bit-strings is $\{0,1\}^*$. The symbol $\oplus$ denotes the bitwise XOR operation of two strings. The symbol 0^n represents the n-bit string consisting of only zeros. Given a *block length* n, concatenation of 10^* to a string means appending a one followed by the minimum number of zeros to make the total string length a multiple of n bits.

Throughout, $\mathbf{P}$ denotes a probability measure. We write $\mathbf{P}\left[\mathsf{A} \;\middle|\; \mathsf{B}\right]$ to denote

the probability of event A given B. By $K \xleftarrow{\$} \mathsf{K}$ we mean that K is chosen uniformly at random from the set K, where K is implicitly assumed to be finite.

We will use the following result throughout the thesis.

Lemma 1. *Say that A and B are independent random variables over a finite group G. If A is uniformly distributed, then $A + B$ is uniformly distributed.*

2.2 Binary Fields

The set $\{0,1\}^n$ of bit strings can be identified with the finite field $\mathrm{GF}(2^n)$ consisting of 2^n elements. The elements of $\mathrm{GF}(2^n)$ can be represented as polynomials of degree less than n over the field $\mathrm{GF}(2)$. The string $a_{n-1}a_{n-2}\cdots a_1a_0 \in \{0,1\}^n$ is then identified with the polynomial $a_{n-1}\mathtt{x}^{n-1} + a_{n-2}\mathtt{x}^{n-2} + \cdots + a_1\mathtt{x} + a_0 \in \mathrm{GF}(2^n)$. Addition in $\{0,1\}^n$ is just addition of polynomials over $\mathrm{GF}(2)$, which is bitwise XOR, $\oplus$. Multiplication is done by fixing an irreducible polynomial $f(\mathtt{x})$ of degree n over the field $\mathrm{GF}(2)$. Given two elements $a(\mathtt{x}), b(\mathtt{x}) \in \mathrm{GF}(2^n)$, their product is defined as $a(\mathtt{x})b(\mathtt{x}) \bmod f(\mathtt{x})$—polynomial multiplication over the field $\mathrm{GF}(2)$ reduced modulo $f(\mathtt{x})$. We simply write $a(\mathtt{x})b(\mathtt{x})$ and $a(\mathtt{x}) \cdot b(\mathtt{x})$ to mean the product in the field $\mathrm{GF}(2^n)$.

The set $\{0,1\}^n$ can be also be identified with the set of integers ranging from 0 through $2^n - 1$: strings $a_{n-1}a_{n-2}\cdots a_1a_0 \in \{0,1\}^n$ are mapped to integers $a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_12 + a_0$. Often elements of $\mathrm{GF}(2^n)$ will be written as integers, by first mapping them to strings, and subsequently to integers. For example, "2" means $\mathtt{x}$, "3" means $\mathtt{x}+1$, and "7" means $\mathtt{x}^2+\mathtt{x}+1$. Multiplications such as $2 \cdot 3$ and 7^2 correspond to those in the field $\mathrm{GF}(2^n)$.

2.3 Algorithms, Adversaries, and Success Measures

Algorithms. We assume the reader generally understands what an "algorithm" is. Throughout the text, we describe stateful, randomized, and deterministic algorithms. A stateful algorithm computes its output based on its input and current state. A randomized algorithm can "flip coins", i.e. generate randomness, each time it is invoked and then use the coins to compute its output. A deterministic, stateless algorithm always returns the same output given the same input.

The *interface* to an algorithm is the set of valid inputs to the algorithm and set of possible outputs the algorithm might make. Interfaces are generally denoted using function notation. For example, an algorithm's interface might

be described as $\mathsf{K} \times \mathsf{M} \to \mathsf{C}$, meaning it accepts inputs from $\mathsf{K} \times \mathsf{M}$ and provides outputs in C.

Adversaries and Oracles. An *adversary* $\mathbf{A}$ is a randomized and stateful algorithm with access to an *oracle* $\mathcal{O}$. An oracle is an algorithm itself, which could represent a cryptographic scheme being analyzed. The interaction between the adversary $\mathbf{A}$ and the oracle $\mathcal{O}$, denoted $\mathbf{A}^{\mathcal{O}}$, generates a *transcript*, which is a sequence of $\mathcal{O}$-inputs, $x_1, x_2, \ldots, x_q$, with corresponding $\mathcal{O}$-outputs, $\mathcal{O}(x_1), \mathcal{O}(x_2), \ldots, \mathcal{O}(x_q)$. The $\mathcal{O}$-inputs x_i are constructed sequentially by the adversary $\mathbf{A}$ using its previously received $\mathcal{O}$-outputs $\mathcal{O}(x_1), \ldots, \mathcal{O}(x_{i-1})$.

Adversarial and oracle interfaces are assumed to be *compatible*, meaning that adversaries always generate oracle inputs which lie in the oracle's input domain. The interfaces of two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ are also said to *match* if the input domains and the output domains of the oracles are the same.

Games. Adversarial success measures are defined in settings called *games*. In *event-based* games, adversaries must trigger an event defined with respect to the transcript generated from the oracle interaction. In this case, adversarial success probability, or the adversary's *advantage*, is measured as the probability the event is satisfied. An example of an event-based game can be found in Section 3.2.

Another game type is *indistinguishability*. Here adversaries are given access to an oracle which could be one of two algorithms. The task of the adversary is to say which of the two algorithms it is interacting with. An example of an indistinguishability game is given in Section 3.1.2. The indistinguishability advantage of adversary $\mathbf{A}$ in distinguishing algorithm f from g is

$$\underset{\mathbf{A}}{\Delta}(f\,;\,g) \stackrel{\text{def}}{=} \left|\mathbf{P}\left[\mathbf{A}^{f}=1\right] - \mathbf{P}\left[\mathbf{A}^{g}=1\right]\right|, \tag{2.1}$$

where the notation $\mathbf{A}^{\mathcal{O}} = 1$ is the event that $\mathbf{A}$ outputs 1 when interacting with oracle $\mathcal{O}$. The probabilities are defined over the probability spaces of $\mathbf{A}$ and $\mathcal{O}$. An adversary which can reliably distinguish between f and g will have indistinguishability advantage close to one. The Δ notation can be generalized to any class of adversaries $\mathbb{A}$ as follows,

$$\underset{\mathbb{A}}{\Delta}(f\,;\,g) \stackrel{\text{def}}{=} \sup_{\mathbf{A}\in\mathbb{A}} \left|\mathbf{P}\left[\mathbf{A}^{f}=1\right] - \mathbf{P}\left[\mathbf{A}^{g}=1\right]\right|, \tag{2.2}$$

which is the supremum of the distinguishing advantages over all adversaries in $\mathbb{A}$.

Multiple oracles are separated by a comma, for example $\Delta(f_1, f_2\,;\, g_1, g_2)$ denotes distinguishing (f_1, f_2) from (g_1, g_2). If $\mathbf{A}$ is distinguishing $(f_1, f_2, \ldots, f_k)$ from $(g_1, g_2, \ldots, g_k)$, then $\mathcal{O}_i$ denotes the ith oracle that $\mathbf{A}$ can access, that is, either f_i or g_i depending upon the oracle sequence it is interacting with. In particular, the order in which the oracles are written is important: $\Delta(f_1, f_2\,;\, g_1, g_2)$ is not the same as $\Delta(f_2, f_1\,;\, g_1, g_2)$. The oracle sequence $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$ can always be identified with the oracle $\mathcal{O}(i, x) \stackrel{\text{def}}{=} \mathcal{O}_i(x)$, hence any statement involving a single oracle can be applied to a sequence of oracles as well.

2.4 Reductions

A systematic approach to investigating an algorithm's security involves looking for attacks, and in the absence thereof, attempting to prove security. Proving that efficient algorithms are secure is generally considered infeasible, therefore the main method of analyzing algorithms is to search for resistance against as many attacks as possible; this approach is usually called *cryptanalysis*.

However, if an algorithm is built using a building block, then one might be able to *reduce* the security of the algorithm in question to some property of the building block; this approach is commonly called the *standard model*, and is the main method of analyzing security in this thesis. Such a reduction converts an adversary attacking the algorithm to an adversary attacking the underlying building block, and if the building block is secure, meaning there are no efficient adversaries attacking it, then, using the reduction, we know there are no adversaries attacking the original algorithm.

The advantage to the standard model is that one can formally reason about why security is preserved without having to resort to relying on the absence of attacks for security. However, the standard model always requires some building block to start with, making cryptanalysis indispensable.

Another approach to reasoning about security is to *idealize* the underlying building blocks, meaning, instead of reducing the algorithm's security to its building block, one replaces the building block with an ideal mathematical object; this is called the *ideal model*. Such an approach is used if there is no obvious theoretical connection between the algorithm's security and any property of the building block; see for example our publications on permutation-based cryptography [11, 105, 129].

Using the ideal model one can no longer claim that an attack against the algorithm can be reduced to an attack against the building block. However, analysis performed in the ideal model still excludes so-called *generic* attacks,

that is, ones which do not use any property of the underlying building block. Despite the lack of a theoretical connection, for practice there does not seem to be an issue in idealizing the building block, assuming the actual building block used does not contain any weaknesses.

2.5 Efficiency

All algorithms and adversaries throughout the text are considered to be "efficient", where picking the right definition of efficiency is outside the scope of the text. See Bernstein and Lange [40] for a discussion on the issues surrounding efficiency of adversaries. The reductions used in the text are also assumed to be efficient, although we do not explicitly measure their efficiency. We list the most commonly used reductions in the text, which should be "efficient" using any reasonable definition.

Definition 2.5.1. Consider an adversary $\mathbf{A}$ interacting with a single oracle. Define $\mathbf{A}(f\circ)$ to be the adversary which interacts with oracle $\mathcal{O}$ as follows: $\mathbf{A}(f\circ)$ runs $\mathbf{A}$ and simulates an oracle for $\mathbf{A}$ by responding to an $\mathbf{A}$-query x via $f(\mathcal{O}(x))$, where f is simulated using $\mathbf{A}(f\circ)$'s own randomness. When $\mathbf{A}$ terminates, $\mathbf{A}(f\circ)$ uses $\mathbf{A}$'s output as its own. Similarly, let $\mathbf{A}(\circ f)$ be the adversary which runs $\mathbf{A}$, simulates $\mathbf{A}$'s oracle queries using $\mathcal{O} \circ f$, and forwards $\mathbf{A}$'s output.

Definition 2.5.2. Let $\mathbf{A}$ be an adversary interacting with two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. Define $\mathbf{A}(f, \cdot)$ to be the adversary interacting with oracle $\mathcal{O}$, which simulates f with its own randomness, runs $\mathbf{A}$, and when $\mathbf{A}$ makes an $\mathcal{O}_1$-query x returns $f(x)$, and returns $\mathcal{O}(x)$ when $\mathbf{A}$ makes an $\mathcal{O}_2$-query x. When $\mathbf{A}$ terminates, $\mathbf{A}(f, \cdot)$ forwards $\mathbf{A}$'s output. Define $\mathbf{A}(\cdot, f)$ similarly.

The above reductions can be combined to create more advanced reductions, such as $\mathbf{A}(\circ f, \cdot)$, which composes f to one oracle, and forwards the second oracle to $\mathbf{A}$.

2.6 Properties of $\triangle$

Let f, g, and h be oracles with matching interfaces, and let $\mathbf{A}$ be an adversary compatible with f.

Proposition 2.6.1.

$$\underset{\mathbf{A}}{\Delta}(f\,;\,g) = \underset{\mathbf{A}}{\Delta}(g\,;\,f) \qquad \textit{(symmetry)} \tag{2.3}$$

$$\underset{\mathbf{A}}{\Delta}(f\,;\,h) \leq \underset{\mathbf{A}}{\Delta}(f\,;\,g) + \underset{\mathbf{A}}{\Delta}(g\,;\,h) \qquad \textit{(triangle inequality)}\,. \tag{2.4}$$

Proof. Both properties follow from the fact that the absolute value is used in the definition of Δ. □

Proposition 2.6.2. *Say that f is independent of g and h, then*

$$\underset{\mathbf{A}}{\Delta}(f \circ g\,;\, f \circ h) \leq \underset{\mathbf{A}(f\circ)}{\Delta}(g\,;\,h) \tag{2.5}$$

$$\underset{\mathbf{A}}{\Delta}(g \circ f\,;\, h \circ f) \leq \underset{\mathbf{A}(\circ f)}{\Delta}(g\,;\,h)\,. \tag{2.6}$$

Proof. Since f is independent of g and h, $\mathbf{A}(f\circ)$ can simulate $f \circ g$ and $f \circ h$ perfectly, which means $\mathbf{A}$'s distinguishing game is simulated perfectly. In particular, if $\mathbf{A}$ succeeds in distinguishing $f \circ g$ from $f \circ h$, then $\mathbf{A}(f\circ)$ succeeds. □

Proposition 2.6.3. *Say that f is independent of g, h and e, and that e is independent of g, h, and f, then*

$$\underset{\mathbf{A}}{\Delta}(f,g\,;\,f,h) \leq \underset{\mathbf{A}(f,\cdot)}{\Delta}(g\,;\,h) \tag{2.7}$$

$$\underset{\mathbf{A}}{\Delta}(f,g\,;\,h,e) \leq \underset{\mathbf{A}(f,\cdot)}{\Delta}(g\,;\,e) + \underset{\mathbf{A}(\cdot,e)}{\Delta}(f\,;\,h)\,. \tag{2.8}$$

The proof is identical to the one for Proposition 2.6.2.

2.7 Ideal Primitives

Often the quality of cryptographic algorithms will be measured with how well they approximate ideal mathematical objects, also called *ideal primitives*. We list some of the most commonly used ideal primitives in the thesis.

1. A *uniformly distributed random function* (URF) from X to Y is a uniformly distributed random variable over the set of all functions from X to Y, where X and Y are assumed to be finite.

2. A *uniformly distributed random permutation* (URP) over X is a uniformly distributed random variable over the set of all permutations on X, where X is assumed to be finite.

3. A *uniformly distributed random beacon* (URB) [123, 147] $\pi : \mathsf{X} \to \mathsf{Y}$ is a family of URFs $\{\pi_i\}_{i \geq 0}$, where $\pi_i : \mathsf{X} \to \mathsf{Y}$ is a URF, and if X is the ith input to π, then $\pi(X) = \pi_i(X)$.

All of the above primitives also have a *length-preserving* variant π operating on domain X^*, where for $X \in \mathsf{X}^*$, $\pi(X) = \pi_{|X|}(X)$, where $\{\pi_i\}_{i \geq 0}$ is a family of primitives with π_i operating on X^i. For example, a length-preserving URB $\pi : \mathsf{X}^* \to \mathsf{Y}^*$ is a family of URBs $\{\pi_i\}_{i \geq 0}$, where $\pi_i : \mathsf{X}^i \to \mathsf{Y}^i$ is a URB, and $\pi(X) = \pi_{|X|}(X)$ for $X \in \mathsf{X}^*$.

Furthermore, all primitives also have a *tweakable* variant π, where given a tweak set A, $\pi(A, \cdot) \stackrel{\text{def}}{=} \pi_A(\cdot)$, where $\{\pi_A\}_{A \in \mathsf{A}}$ is some publicly available primitive family. Tweak-access will usually be denoted with superscripts, so $\pi(A, \cdot) = \pi^A(\cdot)$.

The following result, commonly known as the PRP-PRF switching lemma [35,96], computes the distance between a URP and a URF.

Lemma 2. *Let π be a URP over X and φ a URF from X to X, then for any adversary $\mathbf{A}$ making at most q queries,*

$$\underset{\mathbf{A}}{\Delta}(\pi\,;\,\varphi) \leq \frac{q(q-1)}{2\,|\mathsf{X}|}\,. \tag{2.9}$$

See, for example, Chang and Nandi [60] for a proof.

Chapter 3

Basic Security Definitions

We consider a setting in which two parties wish to communicate securely over a channel where adversaries may intercept, modify, and inject data. Assume both parties share a common secret, a *key*. Two aspects to providing security in this so-called *symmetric-key* setting are considered:

1. data confidentiality, or the extent to which adversaries are not able to determine data content when intercepting, and
2. data integrity, or the extent to which adversaries are not able to modify or inject data without the change being detected by the receiver.

Establishing both data confidentiality and integrity might not lead to sufficient security since other vulnerabilities not captured by the above model might be present, such as inundating the channel to mount denial-of-service attacks, or even leaking the fact that party A is communicating with party B. Providing security against other attacks is beyond the scope of this thesis.

This chapter describes formalizations of data confidentiality and integrity, which consist of three parts: scheme descriptions, adversary descriptions, and adversarial success measures. All three parts combine to describe a security model in which schemes can be tested, and potentially proved, for security.

3.1 Confidentiality

3.1.1 Syntax: Encryption Schemes

In its most basic form, a symmetric-key protocol which attempts to achieve data confidentiality, called an *encryption scheme*, consists of three algorithms:

1. a randomized *key generation* algorithm, which outputs a key $K \in \mathsf{K}$,
2. an encryption algorithm $\mathsf{Enc} : \mathsf{K} \times \mathsf{P} \to \mathsf{C}$, which takes a key $K \in \mathsf{K}$ and a *plaintext* P, to return a *ciphertext* $C \in \mathsf{C}$:

$$\mathsf{Enc}(K, P) = C \quad \text{or} \quad \mathsf{Enc}_K(P) = C\,, \tag{3.1}$$

   and
3. a decryption algorithm $\mathsf{Dec} : \mathsf{K} \times \mathsf{C} \to \mathsf{M}$, which takes a key $K \in \mathsf{K}$ and a ciphertext $C \in \mathsf{C}$ and returns some plaintext $P \in \mathsf{P}$:

$$\mathsf{Dec}(K, C) = P \quad \text{or} \quad \mathsf{Dec}_K(C) = P\,. \tag{3.2}$$

Two parties wishing to communicate confidentially first agree upon a key K using the key generation algorithm, which generally consists of choosing K uniformly at random from K, written as $K \xleftarrow{\$} \mathsf{K}$. Anytime a plaintext P is to be communicated, the sender encrypts P using Enc with key K to produce ciphertext $C = \mathsf{Enc}_K(P)$. The receiver decrypts C using Dec and K to produce P. In order for the communication to work, the encryption scheme must be *correct*, meaning for any key $K \in \mathsf{K}$ and plaintext $P \in \mathsf{P}$, encrypting and then decrypting P always results in P:

$$\mathsf{Dec}_K(\mathsf{Enc}_K(P)) = P.$$

A priori, the encryption and decryption algorithms in encryption schemes can be stateful, randomized, or neither, although we will see that the distinction is important for security.

Example 3.1.1 (One-Time-Pad)**.** One of the simplest examples of a stateful encryption scheme is the *one-time-pad* [171]. Let $\mathsf{K} = \{0,1\}^k$, $\mathsf{P} = \{0,1\}^{\leq p}$, and $\mathsf{C} = \{0,1\}^{\leq c}$. The one-time-pad maintains state representing a bit position in the key, initially set to the first bit. It then takes a plaintext P as input and selects a part of the key, K', of length $|P|$ starting from the bit position it has stored, and then performs a bitwise XOR of the plaintext and key to produce the ciphertext: $C = P \oplus K'$. It then advances the bit position to be past the portion of the key it has used. The decryption algorithm does the same as the encryption algorithm, but uses the ciphertext instead of the plaintext. ◀

3.1.2 Security Definition

A confidentiality definition needs to somehow capture the idea that no information can be extracted about the plaintext given the ciphertext. Goldwasser and Micali [83] approach this by saying that an encryption scheme provides confidentiality if

> whatever is efficiently computable about the [plaintext] given the [ciphertext], is also efficiently computable without the [ciphertext].

Bellare, Desai, Jokipii, and Rogaway [25] discuss several formalizations of the above concept, of which we use *real-or-random* confidentiality.

Real-or-random confidentiality describes adversarial success probability via an indistinguishability game in which adversaries must distinguish the encryption of an input they generate themselves, from the encryption of a randomization of the input. For example, an adversary testing the confidentiality of the one-time-pad would either get access to the one-time-pad itself, or the one-time-pad where the plaintexts are randomized. If the adversary is unable to distinguish the two situations, then it cannot tell whether its plaintexts are actually being encrypted by the one-time-pad, or whether its plaintexts are first converted to nonsense, and then encrypted.

Formally, adversary $\mathbf{A}$'s advantage in breaking an encryption scheme's confidentiality is as follows.

Definition 3.1.2 (Confidentiality)**.** Let $\mathsf{P} = \mathsf{X}^*$, and $\$: \mathsf{P} \to \mathsf{P}$ a length-preserving URB. Then the CPA-advantage of adversary $\mathbf{A}$ against encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is given by

$$\mathsf{CPA}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K \,;\, \mathsf{Enc}_K \circ \$)\,, \tag{3.3}$$

where $K \xleftarrow{\$} \mathsf{K}$.

Randomization of the input is represented via composition with the URB: if the URB gets an input of length ℓ, then its output will be some uniformly distributed random value over all plaintexts of length ℓ. Usually X is defined to be $\{0,1\}$, so that $\mathsf{X}^* = \{0,1\}^*$ is the set of all arbitrary-length strings.

Using this definition, encryption schemes do not need to hide the plaintext length. Consider the one-time-pad again. The encryption algorithm XORs a secret random value to each plaintext, which can be written as $\$'(P) \oplus P$, where $\$'$ is a length-preserving URB independent of the game's URB $\$$. If you pass P

through the game's URB \$, you get

$$\$'(\$(P)) \oplus \$(P)\,, \tag{3.4}$$

which is identically distributed to $\$'(P) \oplus P$ (see Lemma 1). Hence, the one-time-pad provides confidentiality according to the above definition, yet it leaks the plaintext length. In most cases encryption schemes will leak plaintext length, however there are applications where hiding the plaintext length is important; see for example Boldyreva, Degabriele, Paterson, and Stam [50, 53] for a formalization of the setting.

3.1.3 Adversarial Capabilities

Definition 3.1.2 does not correspond exactly to the intuition provided by Goldwasser and Micali, since adversaries are given access to the encryption oracle which means they already know the plaintexts being encrypted. This is called the *chosen plaintext attack* (CPA) scenario, where adversaries may choose plaintexts and see the corresponding ciphertexts. Alternatively, one can consider models in which adversaries are given less power, such as *known plaintext attacks*, where adversaries lose access to the encryption oracle and are given a list of plaintexts with corresponding ciphertexts, or *ciphertext-only attacks*, where adversaries are only given a list of ciphertexts, and the plaintexts are generated randomly according to some distribution.

In some situations the weaker settings might be sufficient, yet there are scenarios in practice in which adversaries are able to inject plaintext during encryption, and then intercept the ciphertext. From an attacker's viewpoint, finding ciphertext-only attacks is very useful, because they can be applied everywhere. But from a designer's viewpoint, it is better to create schemes which are secure against the largest class of attacks possible without sacrificing efficiency, which is why we focus on the CPA scenario.

To this end, we also consider an even stronger setting, in which adversaries are given access to the decryption oracle as well; this might happen if adversaries obtain access to the decryption device, a plausible scenario nowadays given the amount of devices connected to the internet. Such attacks are called *chosen ciphertext attacks* (CCA), with corresponding confidentiality formalization as follows.

Definition 3.1.3 (CCA Confidentiality)**.** Let $\mathsf{P} = \mathsf{X}^*$, and let $\$: \mathsf{P} \to \mathsf{P}$ be a length-preserving URB. Then the CCA-advantage of adversary $\mathbf{A}$ against encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is given by

$$\mathsf{CCA}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K, \mathsf{Dec}_K\,;\, \mathsf{Enc}_K \circ \$, \mathsf{Dec}_K)\,, \tag{3.5}$$

where $K \xleftarrow{\$} \mathsf{K}$, and $\mathbf{A}$ may not use the output of an $\mathcal{O}_1$ query as the input to an $\mathcal{O}_2$ query.

Note the restriction on the adversary's queries: it may not encrypt a plaintext and then decrypt it. Such a query sequence would allow the adversary to trivially distinguish, since encryption scheme correctness requires that the decryption of an encryption must return the original plaintext, which is unlikely to happen when interacting with $(\mathsf{Enc}_K \circ \$, \mathsf{Dec}_K)$.

3.1.4 Leaking Repetition

An aspect of confidentiality which might not be obvious at first, is that repeated plaintexts must result in different ciphertexts. For example, say that a sender repeatedly communicates either a "yes" or "no", and that "yes" always encrypts to the same ciphertext, and so does "no". Then not only will the number of "yes" and "no" plaintexts be leaked, but adversaries can also see when the sender is making different decisions, just based on the ciphertext. Going back to Goldwasser and Micali's intuition, adversaries in such a situation would be able to determine plaintext properties which are impossible to determine without the ciphertext.

Such attacks are captured in the CPA definition as follows. Let P_1 and P_2 be two different plaintexts, and let $\mathcal{O}$ be the adversary's oracle. The adversary first queries $\mathcal{O}(P_1) = C_1$, and then again $\mathcal{O}(P_1) = C_2$. If $C_1 = C_2$, the adversary guesses that it is interacting with Enc_K, and otherwise it guesses $\mathsf{Enc}_K \circ \$$. If Enc_K always outputs the same ciphertext with the same plaintext, then C_1 will always equal C_2 when interacting with just Enc_K, but when interacting with $\mathsf{Enc}_K \circ \$$, the URB will convert P_1 and P_2 into two distinct plaintexts with high probability, which means $C_1 = \mathsf{Enc}_K \circ \$(P_1)$ will most likely not equal $C_2 = \mathsf{Enc}_K \circ \$(P_2)$. Therefore encryption schemes must be either randomized or stateful.

3.2 Integrity

Ensuring integrity concerns two aspects. One is being able to distinguish communication received from the intended sender versus communication received from adversaries. The other, related, aspect is being able to determine when the communication has been modified or tampered with.

Conventional approaches to integrity either limit treatment to schemes which provide no confidentiality, as for example presented by Bellare, Kilian, and Rogaway [28], or only consider schemes which also provide confidentiality, as done by Bellare and Namprempre [32] and Katz and Yung [108]. We merge both approaches into a single definition and abstract away details which would only be necessary to provide confidentiality. Furthermore, we allow for the possibility of multiple verification failures to be output, an issue addressed by Boldyreva, Degabriele, Paterson, and Stam [51].

A symmetric-key protocol attempting to provide integrity we call an *authenticator*, and consists of three algorithms:

1. a randomized *key generation* algorithm, which outputs a key $K \in \mathsf{K}$,
2. a *tagging* algorithm $\mathsf{Tag} : \mathsf{K} \times \mathsf{M} \to \mathsf{C}$, which takes a key $K \in \mathsf{K}$ and a message $M \in \mathsf{M}$, to return an output $C \in \mathsf{C}$:

$$\mathsf{Tag}(K, M) = C \quad \text{or} \quad \mathsf{Tag}_K(M) = C\,, \tag{3.6}$$

   and
3. a *verification* algorithm $\mathsf{Ver} : \mathsf{K} \times \mathsf{C} \to \mathsf{S} \cup \mathsf{F}$, which takes a key $K \in \mathsf{K}$ and an input $C \in \mathsf{C}$ and returns an element of $\mathsf{S} \cup \mathsf{F}$:

$$\mathsf{Ver}(K, C) \in \mathsf{S} \cup \mathsf{F} \quad \text{or} \quad \mathsf{Ver}_K(C) \in \mathsf{S} \cup \mathsf{F}\,. \tag{3.7}$$

   The sets S and F are disjoint, corresponding to the "success" symbols and "failure" symbols, respectively.

Two parties wishing to add integrity to their communication first agree upon a key K using the key generation algorithm. Then, whenever a message M is to be communicated, the sender processes M using Tag with key K to produce output $C = \mathsf{Tag}_K(M)$. The receiver verifies the communication C using Ver; verification succeeds if the Ver output is in S, otherwise verification fails. An authenticator is correct if verification of the tagging algorithm output always succeeds, meaning for all $K \in \mathsf{K}$ and $M \in \mathsf{M}$,

$$\mathsf{Ver}_K(\mathsf{Tag}_K(M)) \in \mathsf{S}\,. \tag{3.8}$$

The goal of an authenticator is to ensure that any input not generated using Tag_K is rejected, that is, without access to K one should not be able to produce an element $C \in \mathsf{C}$ such that $\mathsf{Ver}_K(C) \in \mathsf{S}$. As a result, any communication that is tampered with or new communication that is inserted should be rejected by Ver. These ideas are formalized via the following event-based game.

Definition 3.2.1 (Integrity)**.** Let $K \xleftarrow{\$} \mathsf{K}$. Let $\mathbf{A}$ be an adversary interacting with $(\mathsf{Tag}_K, \mathsf{Ver}_K)$, producing q Tag_K inputs $M_1, M_2, \ldots, M_q$ and v Ver_K inputs $C_1, C_2, \ldots, C_v$. Let C'_i and B_j denote the output of $\mathsf{Tag}_K(M_i)$ and $\mathsf{Ver}_K(C_j)$, respectively. Then the Int advantage of adversary $\mathbf{A}$ is given by

$$\mathsf{Int}(\mathbf{A}) \stackrel{\text{def}}{=} \mathbf{P}\left[\exists j \text{ s.t. } B_j \in \mathsf{S} \text{ and } C_j \neq C'_i \text{ for } i = 1, \ldots, q\right]. \tag{3.9}$$

For full generality we allow F to consist of more than one symbol, however when designing schemes there is little reason to do so. If F consists of a single symbol, say $\perp$, then Int-advantage can be characterized in terms of the indistinguishability game

$$\Delta(\mathsf{Tag}_K, \mathsf{Ver}_K\,;\, \mathsf{Tag}_K, \perp), \tag{3.10}$$

where $\perp$ is an algorithm that always outputs $\perp$ and the adversaries are restricted from using the output of Tag_K as the input to the second oracle. This is because an adversary which is able to construct a forgery will not be able to do so when interacting with $\perp$, and can guess that it is interacting with $(\mathsf{Tag}_K, \mathsf{Ver}_K)$ if it is able to successfully construct the forgery. Conversely, any adversary which is able to distinguish $(\mathsf{Tag}_K, \mathsf{Ver}_K)$ and $(\mathsf{Tag}_K, \perp)$ must force Ver_K to output something other than $\perp$, which is exactly a forgery in the Int-game.

Let $\mathbf{B}\langle\cdot\rangle$ denote the reduction which takes an Int-adversary $\mathbf{A}$ and converts it into indistinguishability adversary $\mathbf{B}\langle\mathbf{A}\rangle$ by running $\mathbf{A}$, responding to $\mathbf{A}$'s oracle requests with its own oracles, and outputting 1 if $\mathbf{A}$ successfully forges, and outputting 0 otherwise. Similarly, let $\mathbf{C}\langle\cdot\rangle$ denote the reduction which takes a distinguisher $\mathbf{A}$ and converts it into Int-adversary $\mathbf{C}\langle\mathbf{A}\rangle$ by running $\mathbf{A}$ using $(\mathsf{Tag}_K, \mathsf{Ver}_K)$.

Proposition 3.2.1. *Let* $(\mathsf{Tag}, \mathsf{Ver})$ *be an authenticator with* $\mathsf{F} = \{\perp\}$*, then for any* Int*-adversary* $\mathbf{A}$

$$\mathsf{Int}(\mathbf{A}) = \underset{\mathbf{B}\langle\mathbf{A}\rangle}{\Delta}(\mathsf{Tag}_K, \mathsf{Ver}_K\,;\, \mathsf{Tag}_K, \perp), \tag{3.11}$$

where $K \xleftarrow{\$} \mathsf{K}$ *and* $\perp$ *is an algorithm which always outputs* $\perp$*. Conversely, for any distinguisher* $\mathbf{A}$*,*

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Tag}_K, \mathsf{Ver}_K\,;\, \mathsf{Tag}_K, \perp) \leq \mathsf{Int}(\mathbf{C}\langle\mathbf{A}\rangle). \tag{3.12}$$

Proof. Since

$$\underset{\mathbf{B}\langle\mathbf{A}\rangle}{\Delta}(\mathsf{Tag}_K, \mathsf{Ver}_K\,;\, \mathsf{Tag}_K, \perp) \stackrel{\text{def}}{=}$$

$$\left|\mathbf{P}\left[\mathbf{B}\langle\mathbf{A}\rangle^{\mathsf{Tag}_K, \mathsf{Ver}_K} = 1\right] - \mathbf{P}\left[\mathbf{B}\langle\mathbf{A}\rangle^{\mathsf{Tag}_K, \perp} = 1\right]\right|, \tag{3.13}$$

and

$$\mathbf{P}\left[\mathbf{B}\langle\mathbf{A}\rangle^{\mathsf{Tag}_K,\mathsf{Ver}_K}=1\right]=\mathbf{P}\left[\mathbf{B}\langle\mathbf{A}\rangle^{\mathsf{Tag}_K,\mathsf{Ver}_K}=1\mid\mathbf{A}\text{ succeeds}\right]\mathbf{P}\left[\mathbf{A}\text{ succeeds}\right] \tag{3.14}$$

$$+\,\mathbf{P}\left[\mathbf{B}\langle\mathbf{A}\rangle^{\mathsf{Tag}_K,\mathsf{Ver}_K}=1\mid\mathbf{A}\text{ fails}\right]\mathbf{P}\left[\mathbf{A}\text{ fails}\right] \tag{3.15}$$

$$=1\cdot\mathsf{Int}(\mathbf{A})+0\cdot\mathbf{P}\left[\mathbf{A}\text{ fails}\right], \tag{3.16}$$

and also

$$\mathbf{P}\left[\mathbf{B}\langle\mathbf{A}\rangle^{\mathsf{Tag}_K,\perp}=1\right]=\mathbf{P}\left[\mathbf{B}\langle\mathbf{A}\rangle^{\mathsf{Tag}_K,\perp}=1\mid\mathbf{A}\text{ succeeds}\right]\mathbf{P}\left[\mathbf{A}\text{ succeeds}\right] \tag{3.17}$$

$$+\,\mathbf{P}\left[\mathbf{B}\langle\mathbf{A}\rangle^{\mathsf{Tag}_K,\perp}=1\mid\mathbf{A}\text{ fails}\right]\mathbf{P}\left[\mathbf{A}\text{ fails}\right] \tag{3.18}$$

$$=1\cdot 0+0\cdot 1\,, \tag{3.19}$$

we have our desired result for the first part.

The second part follows from the fact that if $\mathbf{A}$ succeeds in distinguishing, then it must have constructed a forgery, hence $\mathbf{C}\langle\mathbf{A}\rangle$ succeeds as well, and the distinguishing advantage is at most $\mathsf{Int}(\mathbf{C}\langle\mathbf{A}\rangle)$. □

3.3 Combining Confidentiality and Integrity

In practice, just confidentiality or integrity on their own are often not sufficient for security: not only should data be hidden, but the origin and integrity of the communication must be ensured. Confidentiality provides no integrity since, for example, the one-time-pad has optimal confidentiality, but no integrity: attackers can XOR any value to the ciphertext, and the one-time-pad's decryption would not have any method of detecting the changes. Likewise, schemes which provide integrity do not necessarily provide confidentiality.

Authenticated encryption (AE) schemes target both confidentiality and integrity simultaneously. They take as input a key, message, and so-called *associated data*, which only needs to be checked for integrity. Formally, an AE scheme $(\mathsf{Aenc},\mathsf{Adec})$ is an authenticator where

1. the message space is $\mathsf{M}\stackrel{\text{def}}{=}\mathsf{A}\times\mathsf{P}$, with A the associated data and P the plaintexts,

2. the success symbols are the plaintexts, $\mathsf{S} \stackrel{\text{def}}{=} \mathsf{P}$, and
3. the failure symbols are restricted to one pre-defined error symbol, $\perp$, meaning $\mathsf{F} \stackrel{\text{def}}{=} \{\perp\}$.

We will write $\mathsf{Aenc}_K^A(P)$ for $\mathsf{Aenc}(K, A, P)$. Furthermore, for each $A \in \mathsf{A}$, the AE scheme $(\mathsf{Aenc}, \mathsf{Adec})$ specifies the encryption scheme $(\mathsf{Aenc}^A, \mathsf{Adec})$, which is correct for all $A \in \mathsf{A}$, meaning for all $K \in \mathsf{K}$ and $P \in \mathsf{P}$,

$$\mathsf{Adec}_K(\mathsf{Aenc}_K^A(P)) = P\,. \tag{3.20}$$

Note that Adec does not depend on A, which means that the output of Aenc^A should contain sufficient information so that Adec can reconstruct A. This could be done simply by outputting A itself.

Since AE schemes specify a family of encryption schemes, it makes sense to apply the CPA and CCA security definitions to AE schemes, with the additional detail that adversaries have access to a public family of encryption schemes as opposed to a single scheme.

Definition 3.3.1 (AE CPA Confidentiality)**.** Let $\mathsf{P} = \mathsf{X}^*$, and $\$: \mathsf{P} \to \mathsf{P}$ a length-preserving URB. Then the CPA-advantage of adversary $\mathbf{A}$ against AE scheme $(\mathsf{Aenc}, \mathsf{Adec})$ is given by

$$\mathsf{CPA}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K^{(\cdot)}\,;\, \mathsf{Aenc}_K^{(\cdot)} \circ \$)\,, \tag{3.21}$$

where $K \xleftarrow{\$} \mathsf{K}$, and access to a family member $A \in \mathsf{A}$ is specified by the superscript $(\cdot)$.

Definition 3.3.2 (AE CCA Confidentiality)**.** Let $\mathsf{P} = \mathsf{X}^*$, and let $\$: \mathsf{P} \to \mathsf{P}$ be a length-preserving URB. Then the CCA-advantage of adversary $\mathbf{A}$ against AE scheme $(\mathsf{Aenc}, \mathsf{Adec})$ is given by

$$\mathsf{CCA}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K^{(\cdot)}, \mathsf{Dec}_K\,;\, \mathsf{Enc}_K^{(\cdot)} \circ \$, \mathsf{Dec}_K)\,, \tag{3.22}$$

where $K \xleftarrow{\$} \mathsf{K}$, the superscript $(\cdot)$ has the same meaning as in Definition 3.3.1, and $\mathbf{A}$ may not use the output of an $\mathcal{O}_1^{(\cdot)}$ query as the input to an $\mathcal{O}_2$ query.

Note that an AE scheme with A a singleton set is exactly an encryption scheme, hence Definitions 3.3.1 and 3.3.2 are consistent with Definitions 3.1.2 and 3.1.3.

Since an AE scheme should achieve both confidentiality and integrity, its security must be measured via the definitions already given, namely Int and CCA. In fact,

it turns out that an AE scheme satisfying both Int and CPA will already satisfy CCA, as shown by Bellare and Namprempre [32] and Katz and Yung [107]. We restate the result here, with accompanying proof for completeness.

Theorem 1. *Let* $\mathbf{A}$ *be a* CCA*-adversary with respect to the authenticated encryption scheme* $(\mathsf{Aenc}, \mathsf{Adec})$*, then*

$$\mathsf{CCA}(\mathbf{A}) \leq \mathsf{Int}\Big(\mathbf{A}\Big) + \mathsf{Int}\Big(\mathbf{A}(\circ\$, \cdot)\Big) + \mathsf{CPA}\Big(\mathbf{A}(\cdot, \perp)\Big)\,, \tag{3.23}$$

where \$ *is the URB from the* $(\mathsf{Aenc}, \mathsf{Adec})$ CPA*-definition.*

Proof. Using the definition of CCA, and applying the triangle inequality, we get

$$\begin{aligned}\mathsf{CCA}(\mathbf{A}) = \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K^{(\cdot)}, \mathsf{Adec}_K\,;\, \mathsf{Aenc}_K^{(\cdot)} \circ \$, \mathsf{Adec}_K) \leq \\ \underbrace{\underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K^{(\cdot)}, \mathsf{Adec}_K\,;\, \mathsf{Aenc}_K^{(\cdot)}, \perp)}_{(1)} + \underbrace{\underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K^{(\cdot)}, \perp\,;\, \mathsf{Aenc}_K^{(\cdot)} \circ \$, \perp)}_{(2)} \\ + \underbrace{\underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K^{\cdot} \circ \$, \perp\,;\, \mathsf{Aenc}_K^{(\cdot)} \circ \$, \mathsf{Adec}_K)}_{(3)}\,.\end{aligned} \tag{3.24}$$

By Proposition 3.2.1, term (1) is simply the Int-advantage of $\mathbf{A}$ with respect to $(\mathsf{Aenc}, \mathsf{Adec})$. Similarly, term (3) is equal to the Int-advantage of $\mathbf{A}$ with respect to $(\mathsf{Aenc}_K^{(\cdot)} \circ \$, \mathsf{Adec}_K)$, which is equal to the Int-advantage of $\mathbf{A}(\circ\$, \cdot)$.

By Proposition 2.6.3, term (2) is equal to

$$\underset{\mathbf{A}(\cdot,\perp)}{\Delta}(\mathsf{Aenc}_K^{(\cdot)}\,;\, \mathsf{Aenc}_K^{(\cdot)} \circ \$)\,, \tag{3.25}$$

which is the CPA-advantage of $\mathbf{A}(\cdot, \perp)$ with respect to $(\mathsf{Aenc}, \mathsf{Adec})$. As a result, we have our desired bound. □

Chapter 4

Initial Values

The formalizations provided in Chapter 3 make no explicit reference to the underlying state or randomness of the algorithms. This might be a useful abstraction from the point of view of an end-user sending messages through a texting program, but in practice, it is the implementers who come in contact with cryptography, and who need to ensure that state or randomness is properly maintained. In particular, one could assume that implementers are aware of the subtleties involved in maintaining security, and focus on designing cryptography independently. However, such an assumption might not always hold, especially when an implementer is more concerned with efficiency rather than security.

Another approach is to cater cryptography to the implementers, which was taken by Rogaway [154], who extracted state and randomness into an additional input to the encryption scheme: the IV. Then, the encryption and decryption algorithms can be made deterministic and stateless, and the requirements on state or randomness can be made explicit via the IV input. Although this approach sacrifices generality, it allows one to describe many more scenarios where implementations might fail, as opposed to the more abstract model.

In this chapter we describe the algorithms from Chapter 3 with explicit IVs. Formalization of the security definitions will be done with respect to the real-or-random definitions given in Chapter 3, as opposed to using indistinguishability from *random bits*, to be discussed later. We then look at the *abused IV* setting, where IVs may be repeated, which is where the advantage of the real-or-random over the random bits definitions appears. We consider what happens in the abused IV setting to *online encryption schemes*, which are schemes that can output ciphertext as they receive plaintext. Finally we summarize all security definitions presented so far by showing how they relate to each other.

4.1 Describing Randomness and State with IVs

Each of the schemes introduced in the previous section can be formalized with respect to IVs as follows: all "forward" algorithms, $\mathsf{Enc}, \mathsf{Aenc}$, and Tag, receive an additional input N from the space IV, which parametrizes the algorithms, like associated data for AE schemes.

An IV encryption scheme is a triplet of algorithms, with a key generation algorithm, and a family of *deterministic and stateless* algorithms, $\left\{(\mathsf{Enc}^N, \mathsf{Dec})\right\}_{N\in\mathsf{IV}}$, where for each $N \in \mathsf{IV}$, $(\mathsf{Enc}^N, \mathsf{Dec})$ is an encryption scheme. In particular, the correctness condition states that for all $K \in \mathsf{K}, N \in \mathsf{IV}$, and $P \in \mathsf{P}$,

$$\mathsf{Dec}_K(\mathsf{Enc}_K^N(P)) = P\,. \tag{4.1}$$

Similarly, an IV authenticator is a family of deterministic and stateless authenticators $\left\{(\mathsf{Tag}^N, \mathsf{Ver})\right\}_{N\in\mathsf{IV}}$, and an IV AE scheme is a family of deterministic and stateless AE schemes $\left\{(\mathsf{Aenc}^N, \mathsf{Adec})\right\}_{N\in\mathsf{IV}}$. In the case of AE schemes, syntactically there is no difference between the associated data and the IVs.

Since the encryption and decryption algorithms in an IV encryption scheme are stateless and deterministic, they cannot satisfy the CPA definition, because of the attack explained in Section 3.1.4. The way to get around this is to restrict the adversary's IV input. In the case of schemes which use randomness to provide security, the IV-input must be a uniformly, randomly generated value for each new encryption; we call this the *random IV* setting. For schemes which use state, one could require the IV to be a counter which increments for each encryption. Yet Rogaway [154] noticed that one can create encryption schemes where the only requirement on the IV is that it does not repeat, resulting in a more powerful security definition since adversaries are given more freedom; we call this the *nonce IV* setting. Both the random and nonce IV settings can be considered for authenticators and AE schemes as well.

The formal definitions of CPA and CCA security for the random and nonce IV settings are identical to Definition 3.1.2 and Definition 3.1.3, respectively, except the adversaries are additionally restricted in the IV-input. For the random IV setting, adversaries must always use a uniformly, randomly generated value as IV-input for Enc, and similarly, in the nonce IV setting adversaries must always use unique IVs for each Enc input. There is no restriction on Dec input. We distinguish these definitions by prepending a 'r' or 'n' to indicate the random or nonce IV setting, respectively: $\mathsf{r\text{-}CPA}$, $\mathsf{r\text{-}CCA}$, $\mathsf{r\text{-}Int}$, and $\mathsf{n\text{-}CPA}$, $\mathsf{n\text{-}CCA}$, and $\mathsf{n\text{-}Int}$. Naturally, IV-based schemes can always be measured using the CPA,

CCA, and Int definitions if the schemes are wrapped in a construction which generates the appropriate IV.

When the IV is needed for decryption, it must be communicated somehow between the sender and receiver. Often the IV can be a simple counter, in which case the sender and receiver could be synchronized and the IV does not need to be explicitly communicated. If the sender and receiver cannot be synchronized, then the IV should be able to be communicated in the clear without loss of security. In our definition, communication of the IV is implicitly done via the ciphertext space C, which will be $\mathsf{IV} \times \mathsf{Y}$ for some space Y.

4.2 IV Abuse

An advantage to the IV approach is that one can also explore what happens if the IV requirements are not met. In particular, one can look at the *abused IV* setting, where adversaries may repeat IVs. Such IV repetition can occur in practice, as discussed by Fleischmann, Forler, and Lucks [76]. Examples of IV repetition are flawed implementations [55, 57, 110, 114, 175], bad management of nonces by the user, and backup resets or virtual machine clones when the nonce is stored as a counter.

The abused IV setting was first formalized by Rogaway and Shrimpton [156], who determined that the best possible confidentiality one could hope for if IVs were repeated, was that only the repetition would leak and nothing else. Although they focus on AE schemes, we can consider variants of their definitions for just confidentiality. Their approach is to compare the output of the encryption scheme with a "random bits" oracle, as introduced by Rogaway [154]. Concretely, they define the indistinguishability advantage of an adversary **A** in the abused IV setting via

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}\,;\,\$^{(\cdot)})\,, \tag{4.2}$$

where \$ is a family of URFs with the property that for all N, K, and P, $\left|\$^N(P)\right| = \left|\mathsf{Enc}_K^N(P)\right|$. The advantage to designing schemes with this property is that their outputs will look uniformly random, which is useful for many applications. Yet, as a definition of confidentiality, it does not capture all possible attacks.

In fact, the statement that nothing but equality is leaked can be misleading, and in the abused IV setting there is little security when messages have low entropy. For example, if an adversary knows all but one byte of the plaintext P corresponding to a given ciphertext C, then if it is able to query the 256 potential plaintexts $P_1, P_2, \ldots, P_{256}$ and receive the corresponding ciphertexts

$C_1, C_2, \ldots, C_{256}$, it can determine P by comparing C with C_i for all i. Hence, the abused IV setting *cannot* offer confidentiality.

Nevertheless, the above attack cannot be captured in the random bits definition from Equation (4.2). Furthermore, Rogaway and Shrimpton [156] show that there are schemes which have good bounds relative to Equation (4.2). This would indicate that the random bits definition is not a good measure of confidentiality in the abused IV setting. Instead, we depart from their formalization, and use definitions which stay closer to intuition.

The IV-based CPA and CCA definitions cannot be used directly when IVs are repeated since the plaintexts are randomized using a URB, which always outputs a new random value regardless of the input. However, if the URB is replaced by a tweakable URF with tweak set IV, then repeated IVs will result in the same URF being used, which models the fact that repetition of ciphertexts is allowed, but nothing else besides repetition of plaintexts is leaked.

Definition 4.2.1 (Abused IV CPA)**.** Let $\mathsf{P} = \mathsf{X}^*$ and let $\$: \mathsf{IV} \times \mathsf{P} \to \mathsf{P}$ be a tweaked, length-preserving URF. Then the a-CPA advantage of an adversary $\mathbf{A}$ against encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is given by

$$\mathsf{a\text{-}CPA}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)} ; \mathsf{Enc}_K^{(\cdot)} \circ \$^{(\cdot)}) \,, \tag{4.3}$$

where $K \xleftarrow{\$} \mathsf{K}$, and the superscript $(\cdot)$ indicates that adversaries have direct access to the IV input. Note that the same IV is used for both $\mathsf{Enc}_K^{(\cdot)}$ and $\$^{(\cdot)}$ in the oracle $\mathsf{Enc}_K^{(\cdot)} \circ \$^{(\cdot)}$.

The corresponding CCA definition adds access to Dec_K and prohibits adversaries from using the output of the first oracle as input to Dec_K. Furthermore, note that IV can be extended to include associated data, which means that AE schemes are covered by the definition as well.

The following theorem illustrates the limits that the a-CPA definition imposes on encryption schemes: their confidentiality is low when the encrypted plaintexts are short, and increases relative to the plaintext length.

Theorem 2. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme defined over plaintexts* $\mathsf{P} = \mathsf{X}^*$*, then there exists an* a-CPA*-adversary* $\mathbf{A}$ *making* q *queries of length at least* $\ell \geq 1$ *such that*

$$\mathsf{a\text{-}CPA}(\mathbf{A}) \geq \frac{q^2}{|\mathsf{X}|^{\ell+1}} \,, \tag{4.4}$$

where $q < |\mathsf{X}|^{\ell/2}$.

Proof. The adversary fixes an IV, and makes all queries under the same IV. It then generates q distinct plaintexts $P_1, P_2, \ldots, P_q$ of length ℓ. If $\mathbf{A}$ is interacting with Enc_K, then by injectivity of Enc_K the q P_i get mapped to q different ciphertexts. If $\mathbf{A}$ is interacting with $\mathsf{Enc}_K \circ \$$, then the probability that there is a collision among the $\$(P_i)$ is at least $q^2/\left|\mathsf{X}\right|^{\ell+1}$. If there is such a collision, then two ciphertexts will collide, and $\mathbf{A}$ can distinguish with probability one. □

The above result shows that one must either restrict attention to adversaries which make sufficiently long queries or have X be sufficiently large in order to get meaningful results in the abused IV setting. Such a generic attack is not possible in Rogaway and Shrimpton [156]'s formalization, indicating that a-CPA might lie closer to the intuition behind abused IV security.

Little changes for integrity when IVs are repeated, hence the definition of a-INT is the same as for r-INT and n-INT, but with no restrictions on the adversaries. In fact, it is possible to achieve full integrity in the abused IV setting.

4.3 Online Encryption

Observe that the n-CPA and r-CPA definitions make explicit the fact that Enc must sufficiently "mix" the entire input plaintext P, since the URB outputs independent values for different plaintexts. An important class of highly efficient encryption schemes does not mix the input completely, and relies on random IVs or nonce IVs to provide "fresh" information each time a new plaintext is input. Such schemes are often referred to as *online* encryption schemes, which can encrypt "on-the-fly": as they receive plaintext, they can produce ciphertext nearly immediately without seeing the full plaintext. Many online schemes have been implemented in practice, and it is useful to understand how their security degrades in the abused IV setting.

For example, consider an encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ where

$$\mathsf{Enc}_K(N, P_1P_2) = f_K^1(N, P_1)f_K^2(N, P_1, P_2)\,, \tag{4.5}$$

meaning the ciphertext is made of two parts: one which depends only on N and P_1, and one which depends on everything. Then it cannot satisfy a-CPA because an adversary could distinguish by keeping N and P_1 constant, and querying (N, P_1, P_2) and (N, P_1, P_2') where $P_2' \neq P_2$. If such an adversary is interacting with $\mathsf{Enc}_K^{(\cdot)}$, then it sees that the first part of the ciphertext is the same for both (N, P_1, P_2) and (N, P_1, P_2'), whereas if the adversary is interacting with $\mathsf{Enc}_K^{(\cdot)} \circ \$^{(\cdot)}$, then it is very unlikely that the first part of the ciphertext remains constant because $\$^N(P_1, P_2)$ and $\$^N(P_1, P_2')$ are independent, random values.

From the example it is clear that the a-CPA definition does not allow one to describe online encryption scheme security, since all security is lost regardless of the plaintext length. Instead, a weakening of a-CPA is necessary. By changing $ from a family of length-preserving URFs to one which also preserves prefixes, one can describe a "best possible" security goal for online encryption schemes.

Definition 4.3.1 (Prefix-Preserving URF)**.** A prefix-preserving URF π from X^* to Y^* is a family of URFs $\{\pi_i\}_{i\geq 0}$ with $\pi_i : \mathsf{X}^i \to \mathsf{Y}$, such that

$$\pi(X) = (\pi_1(X_1), \pi_2(X_1, X_2), \ldots, \pi_{|X|}(X_1, \ldots, X_{|X|})) \tag{4.6}$$

for $X \in \mathsf{X}^*$.

Definition 4.3.2 (Online Abused IV)**.** Let $\mathsf{P} = \mathsf{X}^*$, and let $ be a tweakable prefix-preserving URF from P to P with tweak set IV. Then the oa-CPA advantage of an adversary $\mathbf{A}$ against encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is given by

$$\mathsf{oa\text{-}CPA}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}\,;\, \mathsf{Enc}_K^{(\cdot)} \circ \$^{(\cdot)})\,, \tag{4.7}$$

where $K \xleftarrow{\$} \mathsf{K}$ and the superscript $(\cdot)$ indicates that adversaries have direct access to the IV input. Note that the same IV is used for both $\mathsf{Enc}_K^{(\cdot)}$ and $\$^{(\cdot)}$ in the oracle $\mathsf{Enc}_K^{(\cdot)} \circ \$^{(\cdot)}$.

Like encryption schemes, AE schemes can also be online, in which case the above definition also holds. As with a-CCA, oa-CCA adds access to Dec_K with the restriction that outputs of the first oracle cannot be used as inputs to Dec_K.

As is the case with non-online schemes, the abused IV setting guarantees *no* confidentiality. Furthermore, the low-entropy attack from the previous section can be extended to messages for which only a prefix of the message is known to be low-entropy, as described by Hoang, Reyhanitabar, Rogaway, and Vizár [93]. Whereas previous security definitions of online abused IV confidentiality [11, 13, 76] would allow schemes to achieve good advantage, we see that oa-CPA places stronger limits.

Theorem 3. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme defined over plaintexts* $\mathsf{P} = \mathsf{X}^*$*, then there exists an* oa-CPA*-adversary* $\mathbf{A}$ *making* q *queries of length at least* $\ell \geq 1$ *such that*

$$\mathsf{oa\text{-}CPA}(\mathbf{A}) \geq \frac{q^2}{|\mathsf{X}|^2}\,, \tag{4.8}$$

where $q < |\mathsf{X}|^{1/2}$.

Proof. The adversary fixes an IV, and makes all queries under the same IV. It then generates q distinct elements $X_1, X_2, \ldots, X_q \in \mathsf{X}$, and a plaintext P of

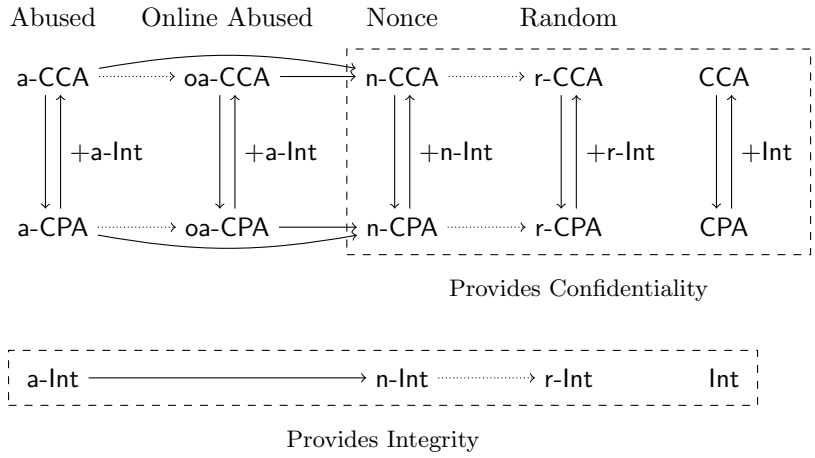


Figure 4.1: Implications between basic security definitions. Dotted arrows mean that there is security loss in the reduction.

length $\ell - 1$. It queries the plaintexts PX_i for $i = 1, \ldots, q$. If $\mathbf{A}$ is interacting with Enc_K, then by injectivity of Enc_K the q PX_i get mapped to q different ciphertexts. If $\mathbf{A}$ is interacting with $\mathsf{Enc}_K \circ \$$, then the probability that there is a collision among the $\$(PX_i)$ is at least $q^2/|\mathsf{X}|^2$, since the first $\ell - 1$ blocks of $\$(PX_i)$ do not change. If there is such a collision, then two ciphertexts will collide, and $\mathbf{A}$ can distinguish with probability one. □

As can be seen by the theorem, the situation for oa-CPA is worse than for a-CPA since the lower bound is independent of the query length that the adversary is forced to make.

4.4 Implications

In this section we show how the security definitions relate to each other, as displayed in Figure 4.1. Note that it does not make sense to compare the non-IV with the IV-based definitions. The definitions which guarantee confidentiality and integrity are indicated, while the remaining definitions indicate "best possible" security when in the given scenarios.

The implications from CCA to CPA security are straightforward, since the reductions just ignore the decryption oracle. The fact that CPA + Int implies CCA was proven in Theorem 1. The proof of Theorem 1 can be extended

to any IV setting, which give all the vertical arrows. The nonce-IV settings directly imply the random-IV settings with a loss of $q^2/|\mathsf{IV}|$ to account for the probability that an IV repeats in the random-IV setting.

The fact that the abused IV confidentiality definitions imply the nonce IV confidentiality definitions is because the $ used in the definition of the abused IV settings is indistinguishable from the $ used in the nonce IV settings as long as the IV is unique. Similarly, the reduction from n-Int to a-Int is immediate. All that remains is proving the connection between the abused and online abused IV settings.

Theorem 4. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme with* $\mathsf{P} = \mathsf{X}^*$*, let* $\$_a$ *denote the randomization function used in the* a-CPA *definition, and* $\$_{oa}$ *the one used in the* oa-CPA *definition, then for any* oa-CPA*-adversary* $\mathbf{A}$ *making at most* q *queries,*

$$\mathsf{oa\text{-}CPA}(\mathbf{A}) \leq \mathsf{a\text{-}CPA}(\mathbf{A}) + \mathsf{a\text{-}CPA}(\mathbf{A}(\circ\$_{oa})) + \frac{q^2}{|\mathsf{X}|}\,. \tag{4.9}$$

Proof. By the triangle inequality,

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}\,;\,\mathsf{Enc}_K^{(\cdot)} \circ \$_{oa}^{(\cdot)}) \leq \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}\,;\,\mathsf{Enc}_K^{(\cdot)} \circ \$_a^{(\cdot)}) \tag{4.10}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)} \circ \$_a^{(\cdot)}\,;\,\mathsf{Enc}_K^{(\cdot)} \circ \$_a^{(\cdot)} \circ \$_{oa}^{(\cdot)}) \tag{4.11}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)} \circ \$_a^{(\cdot)} \circ \$_{oa}^{(\cdot)}\,;\,\mathsf{Enc}_K^{(\cdot)} \circ \$_{oa}^{(\cdot)})\,. \tag{4.12}$$

The first and third terms in the sum are $\mathsf{a\text{-}CPA}(\mathbf{A})$ and $\mathsf{a\text{-}CPA}(\mathbf{A}(\circ\$_{oa}))$, respectively. The second term is bounded above by $\Delta(\$_a\,;\,\$_a \circ \$_{oa})$, which is at most $q^2/|\mathsf{X}|$. □

Similar reasoning establishes the same bound for a-CCA and oa-CCA.

Chapter 5

Building Blocks

In this chapter we present the main tools with which the schemes of Chapter 6 will be constructed. These building blocks say nothing of how to achieve either confidentiality and integrity, and their significance lies in their ability to approximate ideal mathematical objects, even though in some cases only minor modifications are necessary to achieve security.

In order to illustrate how the building blocks could be used in actual constructions, throughout the chapter examples will illustrate how to create higher-level building blocks and schemes which achieve confidentiality and integrity. These constructions will be frequently referred to in Chapter 6. Mixed in with the examples are also two of our constructions, COPE and COBRA, published in Asiacrypt 2013 [13] and FSE 2013 [14].

5.1 Block Ciphers and Modes of Operation

The main tool used in this thesis to achieve confidentiality and integrity is the *block cipher*. A block cipher is a function $\mathsf{E} : \mathsf{K} \times \mathsf{X} \to \mathsf{X}$ where for every key $K \in \mathsf{K}$, $\mathsf{E}(K, \cdot)$ is a permutation with inverse denoted $\mathsf{D}(K, \cdot)$. Usually we will write the keys as subscripts, E_K and D_K. Here the sets K and X are finite, and generally consist of the set of strings of a particular length.

Since block ciphers are used in a wide variety of cryptographic algorithms, they have an equally wide variety of quality measures. The most basic quality measure considers a setting in which the block cipher is keyed with a uniformly random value, and compared with a URP over X. The idea is that the block cipher

allows one to randomly choose a permutation from a small family indexed by keys in K in such a way that the choice is computationally indistinguishable from randomly choosing a permutation over a large set, the set of all permutations.

Definition 5.1.1 (PRP)**.** Let $\mathsf{E} : \mathsf{K} \times \mathsf{X} \to \mathsf{X}$ be a block cipher. Then the pseudorandom permutation (PRP) advantage of adversary $\mathbf{A}$ against E is

$$\mathsf{PRP}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K \,;\, \pi)\,, \tag{5.1}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a URP over X.

In the above definition adversaries are only given access to the "forward" oracle, and not D. The following stronger requirement on the block cipher gives adversaries access to the inverse.

Definition 5.1.2 (SPRP)**.** Let $\mathsf{E} : \mathsf{K} \times \mathsf{X} \to \mathsf{X}$ be a block cipher. Then the strong pseudorandom permutation (SPRP) advantage of adversary $\mathbf{A}$ against E is

$$\mathsf{SPRP}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K, \mathsf{E}_K^{-1} \,;\, \pi, \pi^{-1})\,, \tag{5.2}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a URP on X.

The PRP and SPRP measures on their own say little about how the block cipher can be used to achieve confidentiality and integrity. Furthermore, X is in practice often small. For example, the Advanced Encryption Standard (AES) [67] block cipher only processes strings of length 128 bits.

In order to achieve security, block ciphers are usually used in so-called *modes of operation*, which are constructions that make use of block ciphers as a black box.

Example 5.1.3 (CTR Mode)**.** A simple mode to achieve confidentiality is *counter mode* (CTR). CTR mode uses a block cipher with $\mathsf{X} \stackrel{\text{def}}{=} \{0,1\}^n$ to achieve confidentiality for plaintexts of length up to $2^s \cdot n$ bits, where s is some predefined integer not greater than n.

Given a key $K \in \mathsf{K}$, a plaintext P, and a nonce of length $n - s$ bits, CTR mode divides P into as many complete n-bit blocks as possible $P_1, P_2, \ldots, P_{\ell-1}$, and a final block of length at most n bits, P_ℓ. Then it generates "counter" values $1_s, 2_s, \ldots, \ell_s$, each s bits long, with the property that $i_s \neq j_s$ if $i \neq j$. Each counter value is concatenated with the nonce and used as input to the block cipher to generate the following outputs:

$$X_i \stackrel{\text{def}}{=} \mathsf{E}_K(N \| i_s)\,. \tag{5.3}$$

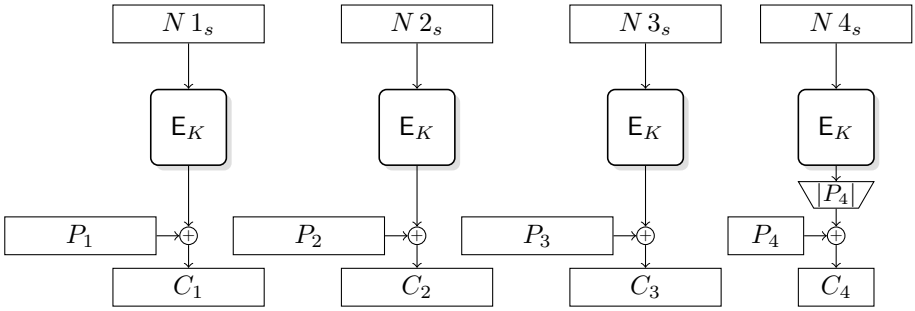


Figure 5.1: CTR mode operating on a 4-block plaintext $P = P_1P_2P_3P_4$, where $|P_4|$ is not necessarily equal to the block size. Truncation to $|P_4|$ bits is indicated with a trapezium.

The resulting sequence of outputs $X_1, X_2, \ldots, X_\ell$ can be viewed as a long key, much like the one-time-pad. Each block X_i is then XORed with the corresponding plaintext block P_i to generate the ciphertext, with the last block X_ℓ appropriately truncated to match the size of P_ℓ. Much like the one-time-pad, decryption is exactly the same as encryption. Figure 5.1 displays a diagram of CTR mode. ◀

Example 5.1.4 (CBC Mode)**.** Another simple mode to achieve confidentiality is the *cipher block chaining* (CBC) mode [139]. Like CTR, it uses a block cipher with $\mathsf{X} \stackrel{\text{def}}{=} \{0,1\}^n$. We describe it for plaintexts which are a concatenation of blocks, $(P_1, P_2, \ldots, P_\ell) \in \mathsf{X}^+$. CBC takes a random IV R and generates block cipher input by XORing the previous block cipher output with the next plaintext block:

$$C_0 = R \tag{5.4}$$

$$C_i = \mathsf{E}_K(P_i \oplus C_{i-1}) \quad \text{for } i = 1, \ldots, \ell\,. \tag{5.5}$$

Decryption reverses the above process:

$$C_0 = R \tag{5.6}$$

$$P_i = \mathsf{D}_K(C_i) \oplus C_{i-1} \quad \text{for } i = 1, \ldots, \ell\,. \tag{5.7}$$

Figure 5.2 depicts CBC mode encryption and decryption. ◀

If the modes use the block cipher inverse, then the block cipher needs to have good SPRP quality, otherwise PRP suffices. For example, CTR mode only uses forward block cipher calls, whereas CBC mode uses both forward and inverse,

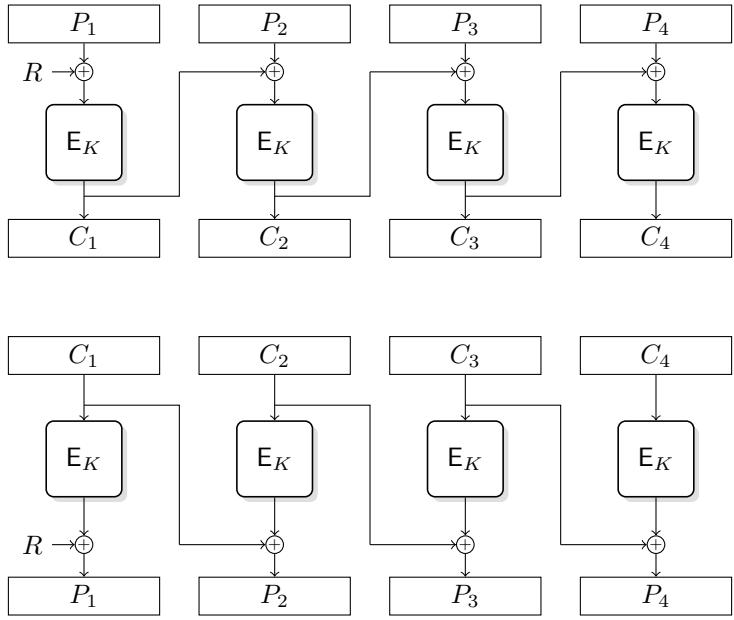


Figure 5.2: CBC mode encryption and decryption for a 4-block plaintext $P = P_1P_2P_3P_4$ and ciphertext $C = C_1C_2C_3C_4$.

hence CTR mode only relies on the PRP quality of a block cipher, whereas CBC mode on the SPRP quality. In some cases just *unpredictability* of the block cipher is necessary, which measures how well adversaries are able to predict the outputs of block ciphers which are not already known, a strictly weaker requirement than PRP. See for example the work done for authenticators [68, 71–73, 124] and on the AE scheme OCB [15].

The way security is proved for modes of operation is by reducing the mode's security to the block cipher's quality. Such a reduction provides a way of converting an attack against the mode into an attack against, for example, the PRP-quality of the block cipher. For all modes in this thesis, the reduction works as follows. Let $\mathbf{A}$ be an adversary attacking the security of the mode. Reduction $\mathbf{B}\langle\cdot\rangle$ attacking the PRP quality of the block cipher is given access to an oracle $\mathcal{O}$, which could either be E_K or π. Adversary $\mathbf{B}\langle\mathbf{A}\rangle$ runs $\mathbf{A}$, and responds to $\mathbf{A}$'s oracle queries by constructing the mode with $\mathcal{O}$. For example, with CTR mode $\mathbf{B}\langle\mathbf{A}\rangle$ would generate the inputs to the block cipher calls and then XOR the output of the resulting $\mathcal{O}$ calls to the plaintext it receives. In general, when referring to a scheme's *mode reduction*, we refer to the construction $\mathbf{B}\langle\cdot\rangle$ corresponding to the given mode.

Using the triangle inequality we get that the mode insecurity with E_K is less than the mode insecurity with π, plus the difference in insecurity between the mode with E_K and the mode with π, or in formula form,

$$\mathsf{E}_K\text{-Mode-Insecurity}(\mathbf{A}) \leq \pi\text{-Mode-Insecurity}(\mathbf{A}) + \underset{\mathbf{B}\langle\mathbf{A}\rangle}{\Delta}(\mathsf{E}_K\text{-Mode}\,;\,\pi\text{-Mode})\,. \tag{5.8}$$

The rightmost term, that is, the comparison of the mode using E_K with the mode using π, is simply the the PRP quality of E_K, hence the mode's insecurity using E_K has been reduced to E_K's PRP quality and the mode insecurity using π.

Note that the mode insecurity has not been perfectly reduced to that of the block cipher using the above argument: computing π-Mode-Insecurity still remains. The majority of the work in arguing that modes provide security relies on computing this last term.

The above argument and the following example can all be found in the paper by Bellare, Desai, Jokipii, and Rogaway [25]

Example 5.1.5 (CTR Mode Reduction)**.** We provide an example of a mode reduction by proving that CTR mode achieves n-CPA confidentiality assuming the underlying block cipher is a good PRP and CTR mode using a URP is secure.

Theorem 5. *Let* $(\mathsf{Enc}[\mathsf{E}], \mathsf{Dec}[\mathsf{E}])$ *denote CTR mode with block cipher* E*. Then for any* n-CPA*-adversary* $\mathbf{A}$ *against* $(\mathsf{Enc}[\mathsf{E}], \mathsf{Dec}[\mathsf{E}])$*,*

$$\mathsf{n\text{-}CPA}_{(\mathsf{Enc}[\mathsf{E}],\mathsf{Dec}[\mathsf{E}])}(\mathbf{A}) \leq \mathsf{PRP}(\mathbf{B}\langle\mathbf{A}\rangle) + \mathsf{PRP}(\mathbf{B}\langle\mathbf{A}\rangle(\circ\$)) + \mathsf{n\text{-}CPA}_{(\mathsf{Enc}[\pi],\mathsf{Dec}[\pi])}(\mathbf{A})\,, \tag{5.9}$$

where $\mathbf{B}\langle\cdot\rangle$ *is the CTR mode reduction.*

The above theorem allows one to focus on the n-CPA-advantage of $\mathbf{A}$ against $(\mathsf{Enc}[\pi], \mathsf{Dec}[\pi])$, that is, CTR mode using a URP.

Proof. The triangle inequality in this case can be written as follows:

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\mathsf{E}_K]^{(\cdot)}\,;\,\mathsf{Enc}[\mathsf{E}_K]^{(\cdot)} \circ \$^{(\cdot)}) \leq \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\mathsf{E}_K]^{(\cdot)}\,;\,\mathsf{Enc}[\pi]^{(\cdot)}) \tag{5.10}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\pi]^{(\cdot)}\,;\,\mathsf{Enc}[\pi]^{(\cdot)} \circ \$^{(\cdot)}) \tag{5.11}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\pi]^{(\cdot)} \circ \$^{(\cdot)}\,;\,\mathsf{Enc}[\mathsf{E}_K]^{(\cdot)} \circ \$^{(\cdot)})\,. \tag{5.12}$$

The first and third terms are the (E_K-Mode vs. π-Mode) term from Equation (5.8). Writing $\mathbf{B}$ as shorthand for $\mathbf{B}\langle\mathbf{A}\rangle$, we get

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\mathsf{E}_K]^{(\cdot)}\,;\,\mathsf{Enc}[\pi]^{(\cdot)}) \le \underset{\mathbf{B}}{\Delta}(\mathsf{E}_K\,;\,\pi) = \mathsf{PRP}(\mathbf{B}) \tag{5.13}$$

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\pi]^{(\cdot)}\circ\$^{(\cdot)}\,;\,\mathsf{Enc}[\mathsf{E}_K]^{(\cdot)}\circ\$^{(\cdot)}) \le \underset{\mathbf{B}(\circ\$)}{\Delta}(\pi\,;\,\mathsf{E}_K) = \mathsf{PRP}(\mathbf{B}(\circ\$))\,. \tag{5.14}$$

As a result,

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\mathsf{E}_K]^{(\cdot)}\,;\,\mathsf{Enc}[\mathsf{E}_K]^{(\cdot)}\circ\$^{(\cdot)}) \le \mathsf{PRP}(\mathbf{B}) + \mathsf{PRP}(\mathbf{B}(\circ\$)) \tag{5.15}$$

$$+\,\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\pi]^{(\cdot)}\,;\,\mathsf{Enc}[\pi]^{(\cdot)}\circ\$^{(\cdot)})\,, \tag{5.16}$$

where the last term is the n-CPA-advantage of $\mathbf{A}$ versus $(\mathsf{Enc}[\pi],\mathsf{Dec}[\pi])$. □

Computing the n-CPA of CTR mode with URP π is trivial. The following theorem combined with the previous one complete the reduction of CTR mode's n-CPA bound to the PRP bound of the underlying block cipher, with a loss in reduction of $\sigma^2/2^n$.

Theorem 6. *Let* $(\mathsf{Enc}[\pi],\mathsf{Dec}[\pi])$ *denote CTR mode with URP* π*. Then for any* n-CPA*-adversary* $\mathbf{A}$ *against* $(\mathsf{Enc},\mathsf{Dec})$ *querying at most* σ *blocks of plaintext,*

$$\mathsf{n\text{-}CPA}(\mathbf{A}) \le \frac{\sigma^2}{2^n}\,. \tag{5.17}$$

Proof. Let ρ be a URF from X to X, then

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\pi]^{(\cdot)}\,;\,\mathsf{Enc}[\pi]^{(\cdot)}\circ\$^{(\cdot)}) \le \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\pi]^{(\cdot)}\,;\,\mathsf{Enc}[\rho]^{(\cdot)}) \tag{5.18}$$

$$+\,\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\rho]^{(\cdot)}\,;\,\mathsf{Enc}[\rho]^{(\cdot)}\circ\$^{(\cdot)}) \tag{5.19}$$

$$+\,\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\rho]^{(\cdot)}\circ\$^{(\cdot)}\,;\,\mathsf{Enc}[\pi]^{(\cdot)}\circ\$^{(\cdot)}) \tag{5.20}$$

$$\le 2\cdot\frac{\sigma^2}{2^{n+1}} + \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\rho]^{(\cdot)}\,;\,\mathsf{Enc}[\rho]^{(\cdot)}\circ\$^{(\cdot)})\,, \tag{5.21}$$

where the last inequality follows from Lemma 2. Using a URF ρ, CTR mode always outputs independent, uniformly distributed values, regardless of what its input is, or in other words

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}[\rho]^{(\cdot)}\,;\,\mathsf{Enc}[\rho]^{(\cdot)}\circ\$^{(\cdot)}) = 0\,. \tag{5.22}$$

□

◀

5.2 Tweakable Block Ciphers

A useful generalization of block ciphers is *tweakable block ciphers* [116]. A tweakable block cipher is a function $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{X} \to \mathsf{X}$ where $\mathsf{E}_K(A, \cdot) = \mathsf{E}_K^A(\cdot)$ is a permutation with inverse $\mathsf{D}_K(A, \cdot) = \mathsf{D}_K^A(\cdot)$ for all $K \in \mathsf{K}$ and $A \in \mathsf{A}$. Here X is finite, and A is the set of *tweaks*, which might consist of variable-length strings.

Whereas block ciphers only give access to a single permutation per key, tweakable block ciphers give access to an entire family, with the requirement that each member of the family looks uniform and independent of all other members. Therefore, the idealization of a tweakable block cipher is a tweakable URP, with tweaks from A. Formally, the quality of a tweakable block cipher is measured as follows.

Definition 5.2.1 (PRP for Tweakable Block Ciphers)**.** Let $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{X} \to \mathsf{X}$ be a tweakable block cipher. Then the pseudorandom permutation (PRP) advantage of adversary $\mathbf{A}$ against E is

$$\mathsf{PRP}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K^{(\cdot)} \,;\, \pi^{(\cdot)}) \,, \tag{5.23}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a tweakable URP with tweak set A.

As with block ciphers, the adversaries can also gain access to the inverse operation, resulting in a stronger quality requirement. We denote access to the inverse permutations via $\mathsf{D}_K^{(\cdot)}$ and $\pi^{-1(\cdot)}$.

Definition 5.2.2 (SPRP for Tweakable Block Ciphers)**.** Let $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{X} \to \mathsf{X}$ be a tweakable block cipher. Then the strong pseudorandom permutation (SPRP) advantage of adversary $\mathbf{A}$ against E is

$$\mathsf{SPRP}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K^{(\cdot)}, \mathsf{D}_K^{(\cdot)} \,;\, \pi^{(\cdot)}, \pi^{-1(\cdot)}) \,, \tag{5.24}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a tweakable URP with tweak set A.

The above definitions are consistent with Definitions 5.1.1 and 5.1.2 since a block cipher can be viewed as a tweakable block cipher with a single tweak. Furthermore, modes of operation for tweakable block ciphers are analogous to modes of operation for block ciphers.

Example 5.2.3 (Simplified OCB)**.** A simple confidentiality mode for tweakable block ciphers is the encryption scheme underlying OCB [111, 153, 155], which is an AE scheme. We describe a simplified version of it here.

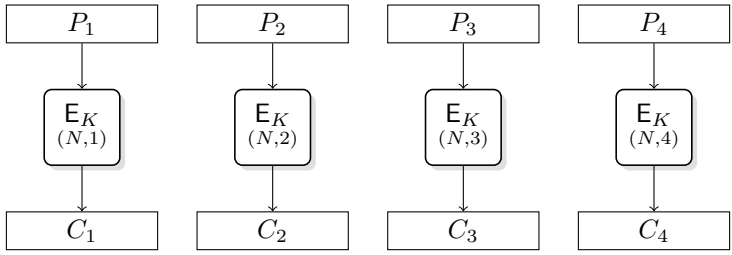


Figure 5.3: Simplified OCB encryption on a plaintext $P = (P_1, P_2, P_3, P_4)$. The tweak corresponding to the tweakable block cipher call is written under E_K.

The simplified OCB encryption scheme uses a tweakable block cipher $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{X} \to \mathsf{X}$, where $\mathsf{A} \stackrel{\text{def}}{=} \mathsf{IV} \times \mathbb{N}$, and operates on plaintexts of the form $\mathsf{P} \stackrel{\text{def}}{=} \mathsf{X}^+$. Given a nonce $N \in \mathsf{IV}$ and a plaintext $P = (P_1, P_2, \ldots, P_\ell)$ the resulting ciphertext is

$$C_i = \mathsf{E}_K^{(N,i)}(P_i) \qquad \text{for } i = 1, \ldots, \ell\,. \tag{5.25}$$

Figure 5.3 depicts a diagram of the process.

With the abstraction to tweakable block ciphers, the argument for why this mode provides confidentiality becomes very simple: each block of plaintext is given its own tweak since nonces are not repeated, and the plaintext is therefore encrypted using an independent, uniformly generated permutation. As a result, the ciphertext blocks will be uniformly distributed and independent of each other. ◀

Examples of tweakable block cipher primitives are the Threefish cipher [75], the TWEAKEY framework [101], and the Hasty Pudding Cipher [161]. An alternative to using the primitives, is to build a tweakable block cipher using a block cipher. Popular methods of turning a block cipher into a tweakable block cipher are XE and XEX by Rogaway [153].

Example 5.2.4 (XE and XEX [153])**.** Let $\mathsf{X} = \{0,1\}^n$. Given a block cipher $\mathsf{E} : \mathsf{K} \times \mathsf{X} \to \mathsf{X}$ and a secret mask $\Delta \in \mathsf{X}$, define

$$\mathsf{E}^1_{K,\Delta}(X) \stackrel{\text{def}}{=} \mathsf{E}_K(X \oplus \Delta) \tag{5.26}$$

$$\mathsf{E}^2_{K,\Delta}(X) \stackrel{\text{def}}{=} \mathsf{E}_K(X \oplus \Delta) \oplus \Delta\,. \tag{5.27}$$

As long as Δ is nonzero, $\mathsf{E}^1_{K,\Delta}$ and $\mathsf{E}^2_{K,\Delta}$ will behave roughly independently of E_K, assuming adversaries may only make forward queries to E^1. Consider a set

of secret masks $\{\Delta_i\}_{i\in\mathsf{A}}$, with A the set of tweaks. Then define the tweakable block ciphers $\mathsf{XE} : \mathsf{K} \times \mathsf{A} \times \mathsf{X} \to \mathsf{X}$ and $\mathsf{XEX} : \mathsf{K} \times \mathsf{A} \times \mathsf{X} \to \mathsf{X}$ by setting

$$\mathsf{XE}_K^A(X) \stackrel{\text{def}}{=} \mathsf{E}^1_{K,\Delta_A}(X)\,, \tag{5.28}$$

and

$$\mathsf{XEX}_K^A(X) \stackrel{\text{def}}{=} \mathsf{E}^2_{K,\Delta_A}(X)\,. \tag{5.29}$$

The doubling method [153] provides a way to produce many different masks Δ from a single secret value $L \stackrel{\text{def}}{=} \mathsf{E}_K(0)$. Identifying X with $\mathrm{GF}(2^n)$ as described in the preliminaries (Chapter 2), the masks are produced as

$$\Delta_{\alpha,\beta,\gamma} = 2^\alpha 3^\beta 7^\gamma \cdot L\,. \tag{5.30}$$

In order to maximize the number of indices α, β, and γ such that Δ is distinct, the irreducible polynomial $f(\mathtt{x})$ needs to be chosen carefully. First, $f(\mathtt{x})$ needs to be primitive, meaning that 2 generates the whole multiplicative group of X. Second, $\log_2 3$ and $\log_2 7$ must both be large. Third, $\log_2 3$ and $\log_2 7$ should be "apart enough" (modulo $2^n - 1$). These conditions ensure that the values $2^\alpha 3^\beta 7^\gamma$ do not collide or become equal to 1, a property needed for security with the XEX. For example, when $n = 128$, the irreducible polynomial $f(\mathtt{x}) = \mathtt{x}^{128} + \mathtt{x}^7 + \mathtt{x}^2 + \mathtt{x} + 1$ satisfies these requirements, making the values $2^\alpha 3^\beta 7^\gamma$ all distinct and not equal to 1 for $\alpha \in [-2^{108}, 2^{108}]$ and $\beta, \gamma \in [-2^7, 2^7]$, except for $(\alpha, \beta, \gamma) = (0, 0, 0)$.

As long as the secret masks are distinct, XE and XEX have reasonably good PRP and SPRP quality. As shown by Rogaway [153], the PRP advantage of XE is bounded above by the PRP advantage of E plus $4.5q^2/2^n$, where q is the number of queries made by the adversary. Similarly, XEX's SPRP advantage is upper bounded by the SPRP advantage of E plus $9.5q^2/2^n$.

◀

5.3 Variable Length Tweakable Ciphers

Both block cipher and tweakable block cipher primitives have the disadvantage that they generally operate on small sets X, such as the set of 128 bit strings. The corresponding objects which operate on much larger sets are called *ciphers* and *tweakable ciphers*.

Let $\mathsf{P} = \mathsf{X}^*$. A tweakable cipher is a function $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{P} \to \mathsf{P}$ where $\mathsf{E}_K(A, \cdot) = \mathsf{E}_K^A(\cdot)$ is a permutation with inverse $\mathsf{D}_K(A, \cdot) = \mathsf{D}_K^A(\cdot)$ for all $K \in \mathsf{K}$

and $A \in \mathsf{A}$. We furthermore require that E preserves plaintext length, meaning $\left|\mathsf{E}_K^A(P)\right| = |P|$. A cipher is a tweakable cipher with a single tweak. As a result, all results and definitions on tweakable ciphers can be applied to ciphers as well.

The quality of tweakable ciphers is measured in the same way as tweakable block ciphers. Both the PRP and SPRP definitions can be applied directly to tweakable ciphers, with π modified to be a tweakable, length-preserving URP.

Definition 5.3.1 (PRP for Tweakable Ciphers)**.** Let $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{P} \to \mathsf{P}$ be a tweakable cipher. Then the pseudorandom permutation (PRP) advantage of adversary $\mathbf{A}$ against E is

$$\mathsf{PRP}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K^{(\cdot)} ; \pi^{(\cdot)}) , \tag{5.31}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a tweaked, length-preserving URP with tweak space A.

Definition 5.3.2 (SPRP for Tweakable Ciphers)**.** Let $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{P} \to \mathsf{P}$ be a tweakable cipher. Then the strong pseudorandom permutation (SPRP) advantage of adversary $\mathbf{A}$ against E is

$$\mathsf{SPRP}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K^{(\cdot)}, \mathsf{D}_K^{(\cdot)} ; \pi^{(\cdot)}, \pi^{-1(\cdot)}) , \tag{5.32}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a tweaked, length-preserving URP with tweak space A.

As we will see in the coming sections, tweakable ciphers are very robust objects, and can provide confidentiality and integrity via simple modifications. But tweakable ciphers are rarely constructed as primitives, and are instead defined as modes of operation for block ciphers or tweakable block ciphers. At least two layers of block cipher calls are necessary in order to construct a tweakable cipher. Examples of tweakable ciphers are the TCT constructions [165], the mode underlying AEZ [92], and Fmix [43].

5.4 Online Ciphers

The downside to tweakable ciphers is that they must mix the entire plaintext sufficiently in order to make every bit of ciphertext depend on every bit of plaintext. This requires internal state which is large enough to store data which is approximately the size of the plaintext, for example, a plaintext which is 1024 bits long will require state that can fit at least 1024 bits. To alleviate the internal state requirement, weaker ciphers can be used, namely *online ciphers* [24] and *tweakable online ciphers*.

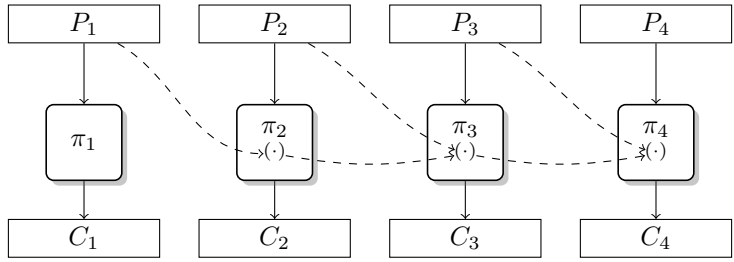


Figure 5.4: Illustration of prefix-preserving URPs. For the inverse, reverse the solid arrows.

A tweakable online cipher is a tweakable cipher where the first ℓ blocks of ciphertext only depend on the first ℓ blocks of plaintext, that is

$$\lfloor \mathsf{E}_K^A(X_1 X_2) \rfloor_\ell = \mathsf{E}_K^A(X_1)\,, \tag{5.33}$$

where $X_1, X_2 \in \mathsf{X}^*$ and $|X_1| = \ell$. As a result, tweakable online ciphers cannot satisfy the PRP and SPRP definitions: when querying two two-block messages (X_1, X_2) and (X_1, X_2') to an online cipher, the resulting outputs will have the same prefix, which is not the case for length-preserving URPs. Instead, tweakable online ciphers are compared with tweakable prefix-preserving URPs.

Definition 5.4.1 (Prefix-Preserving URP)**.** A prefix-preserving URP π on X^* is a family of independent, tweakable URPs $\{\pi_i\}_{i\geq 0}$ with $\pi_i : \mathsf{X}^{i-1} \times \mathsf{X} \to \mathsf{X}$ a URP on X with tweak set X^{i-1}, such that

$$\pi(X) = (\pi_1(X_1), \pi_2^{X_1}(X_2), \ldots, \pi_\ell^{X_1,\ldots,X_{\ell-1}}(X_\ell))\,, \tag{5.34}$$

where $X \in \mathsf{X}^*$ and $|X| = \ell$.

Definition 5.4.2 (Online PRP)**.** Let $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{P} \to \mathsf{P}$ be a tweakable cipher. Then the online pseudorandom permutation (o-PRP) advantage of adversary $\mathbf{A}$ against E is

$$\text{o-PRP}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K^{(\cdot)}\,;\, \pi^{(\cdot)})\,, \tag{5.35}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a tweakable, prefix-preserving URP with tweak space A.

Definition 5.4.3 (Online SPRP)**.** Let $\mathsf{E} : \mathsf{K} \times \mathsf{A} \times \mathsf{X} \to \mathsf{X}$ be a tweakable cipher. Then the online strong pseudorandom permutation (o-SPRP) advantage of adversary $\mathbf{A}$ against E is

$$\text{o-SPRP}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K^{(\cdot)}, \mathsf{D}_K^{(\cdot)}\,;\, \pi^{(\cdot)}, \pi^{-1(\cdot)})\,, \tag{5.36}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a tweakable, prefix-preserving URP with tweak space A.

Since tweakable online ciphers are able to process plaintext and output ciphertext which only depend on preceding plaintext blocks, they will often have internal state which is a fixed amount regardless of the plaintext length, and significantly smaller than with tweakable ciphers. Nevertheless, by composing online ciphers and re-introducing sufficient mixing between the online cipher calls, ciphers can be constructed [10, 44].

Example 5.4.4. An example of an online cipher is TC3 [158], which is given as a mode of operation for a tweakable block cipher $\mathsf{E} : \mathsf{K} \times \mathsf{V} \times \mathsf{X} \to \mathsf{X}$, with $\mathsf{X} = \{0,1\}^n$. We describe the tweakable variant by Fleischmann et al. [76, 77], operating on plaintexts $\mathsf{P} = \mathsf{X}^*$ and tweaks $\mathsf{A} = \mathsf{X}^*$, for some set X. First, the tweak is processed to produce values V_i as follows:

$$V_0 = 0^n \tag{5.37}$$

$$V_i = \mathsf{E}_K^{V_{i-1}}(A_{i-1}) \oplus A_{i-1} \quad \text{for } i = 1, \ldots, \ell \tag{5.38}$$

The remaining plaintext is processed similarly:

$$C_i = \mathsf{E}_K^{V_{\ell+i-1}}(P_i) \quad \text{for } i = 1, 2, \ldots \tag{5.39}$$

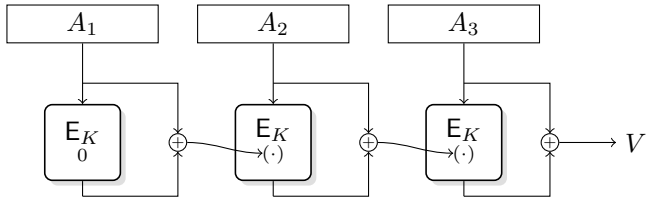
$$V_{\ell+i} = C_i \oplus P_i\,. \tag{5.40}$$

An illustration of tweakable TC3 can be found in Figure 5.5. Rogaway and Zhang [158] prove that TC3 with tweaks is o-SPRP with bound $1.5\sigma^2/2^n$, with σ an upper bound on the number of blocks the adversary queries. Fleischmann et al. [76, 77] prove similar bounds for the tweakable extension. ◀

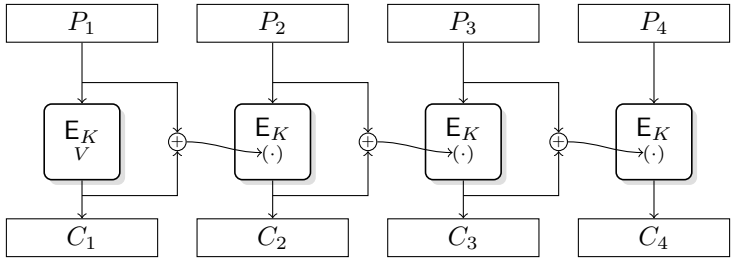
The issue with TC3 is that it is inherently serial: in order to process a plaintext block, the outputs of the previous tweakable block cipher calls are needed. Many online ciphers suffer from similar limitations, with the exceptions being COPE [13], the cipher underlying COBRA [14], and POE [3]. We introduce COPE in this section, and COBRA in the next.

Example 5.4.5. COPE was first introduced as an online cipher. Here we take elements from its counterpart COPA to create the tweakable version of COPE. COPE is illustrated in Figure 5.6 as a mode of operation for tweakable block ciphers; in the original paper the XE and XEX constructions are used to create the tweakable block cipher. The tweaks to the block cipher calls can be split into four different classes: those used to process intermediate tweak values, $(\cdot, 1)$, final tweak values, $(\cdot, 2)$, a first pass over the plaintext, $(\cdot, 3)$, and a second pass over the plaintext, $(\cdot, 4)$.

The tweakable block cipher calls can be called in parallel per layer. Although COPE uses two tweakable block cipher calls per plaintext block versus TC3's



(a) Processing the tweak.



(b) Processing the plaintext.

Figure 5.5: The TC3 online cipher with modification by Fleischmann et al. [76, 77]. Tweaks are written underneath E_K. Tweaks that depend on previous outputs are written $(\cdot)$.

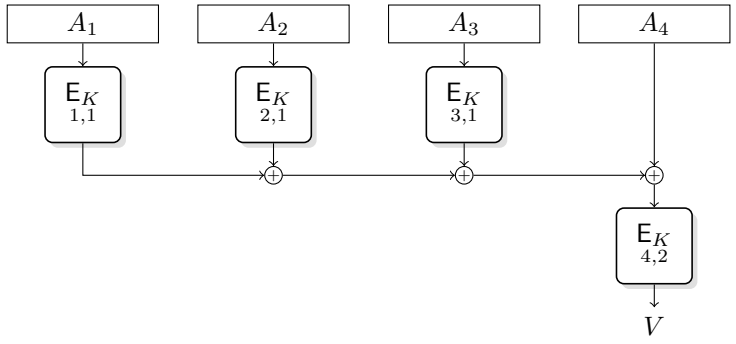
single call, the tweaks used in COPE only depend on the plaintext block position, and can therefore be precomputed, making each tweakable block cipher call significantly cheaper. ◀

5.5 Universal Hash Functions

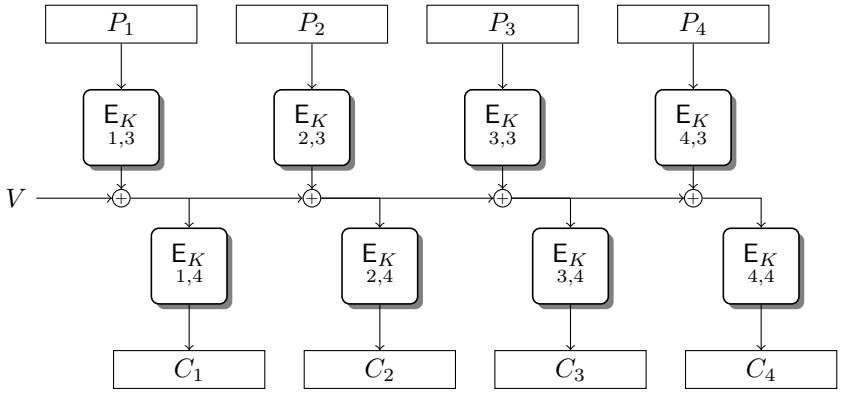
All ciphers described in the previous sections preserve input length and provide an inverse operation. Sometimes the inverse operation is not necessary, and compression is more important. A commonly used tool to compress data is the *universal hash function*, $\mathsf{F} : \mathsf{K} \times \mathsf{M} \to \mathsf{Y}$, which takes keys in K and messages in M to produce outputs in Y. The most important property characterizing a universal hash function is its collision resistance, which is measured via the following definitions.

Definition 5.5.1 (Collision Bound)**.** The collision bound of a keyed function $\mathsf{F} : \mathsf{K} \times \mathsf{M} \to \mathsf{Y}$ is

$$\mathsf{CB}_{\mathsf{F}} \stackrel{\text{def}}{=} \max_{M \neq M'} \mathbf{P}\left[\mathsf{F}(M) = \mathsf{F}(M')\right] . \tag{5.41}$$



(a) Processing the tweak.



(b) Processing plaintext.

Figure 5.6: Tweakable online cipher COPE.

The following definition places a stronger collision resistance requirement on the universal hash function.

Definition 5.5.2 (Additive Collision Bound)**.** Let Y be a group with operation $+$ and let $\mathsf{F} : \mathsf{K} \times \mathsf{M} \to \mathsf{Y}$ be a keyed function. The additive collision bound of F is

$$\mathsf{ACB}_{\mathsf{F}} \stackrel{\text{def}}{=} \max_{M \neq M', Y \in \mathsf{Y}} \mathbf{P}\left[\mathsf{F}(M) = \mathsf{F}(M') + Y\right] . \tag{5.42}$$

Example 5.5.3. Say that $\mathsf{M} = \mathsf{X}^{\leq \ell}$ and $\mathsf{K} = \mathsf{X}$ with X a finite field, then one can construct a universal hash function $\mathsf{F} : \mathsf{K} \times \mathsf{M} \to \mathsf{X}$ by mapping a message $(M_1, M_2, \ldots, M_\ell) \in \mathsf{M}$ and key $K \in \mathsf{K}$ to the value

$$M_1 K^\ell + M_2 K^{\ell-1} + \cdots + M_\ell K \,, \tag{5.43}$$

which is a polynomial in K. The probability that $\mathsf{F}(M) = \mathsf{F}(M') + Y$ is the probability that

$$(M_1 - M'_1)K^\ell + (M_2 - M'_2)K^{\ell-1} + \cdots + (M_\ell - M'_\ell)K - Y = 0\,, \tag{5.44}$$

where $M = (M_1, M_2, \ldots, M_\ell)$ and $M' = (M'_1, M'_2, \ldots, M'_\ell)$. Since the above is a polynomial with degree at most ℓ, there are at most ℓ solutions in K satisfying the above equation, hence the probability of a collision is at most $\ell / |\mathsf{K}|$, which establishes that

$$\mathsf{ACB}_\mathsf{F} \leq \frac{\ell}{|\mathsf{K}|}\,. \tag{5.45}$$

◀

Polynomial-based universal hash functions are often used in practice. Examples include poly1305 [39] and GHASH [125]. The COBRA [14] online cipher uses polynomial-based hashing to create dependency upon preceding plaintext blocks.

Example 5.5.4. The COBRA cipher uses one finite field multiplication and one tweakable block cipher call per plaintext block. COBRA is depicted in Figure 5.7. COBRA replaces COPE's parallelization procedure with a two-round *Feistel structure* in order to avoid use of the inverse block cipher call. Using functions F_1 and F_2 from X to X with $\mathsf{X} = \{0,1\}^n$, the Feistel structure generates an invertible mapping from X^2 to X^2, two rounds of which operate as follows:

$$Y_1 = \mathsf{F}_1(X_1) \oplus X_2 \tag{5.46}$$

$$Y_2 = \mathsf{F}_2(Y_1) \oplus X_1\,, \tag{5.47}$$

with the output being $(Y_1, Y_2) \in \mathsf{X}^2$. The inverse of the operation does not require the inverse of F_1 or F_2:

$$X_1 = \mathsf{F}_2(Y_1) \oplus Y_2 \tag{5.48}$$

$$X_2 = \mathsf{F}_1(X_1) \oplus Y_1\,. \tag{5.49}$$

When considered together, the finite field multiplications form a polynomial-based hash function. By preventing collisions, the universal hash in a sense "tweaks" the tweakable block cipher calls in order to create dependency upon preceding plaintext blocks.

◀

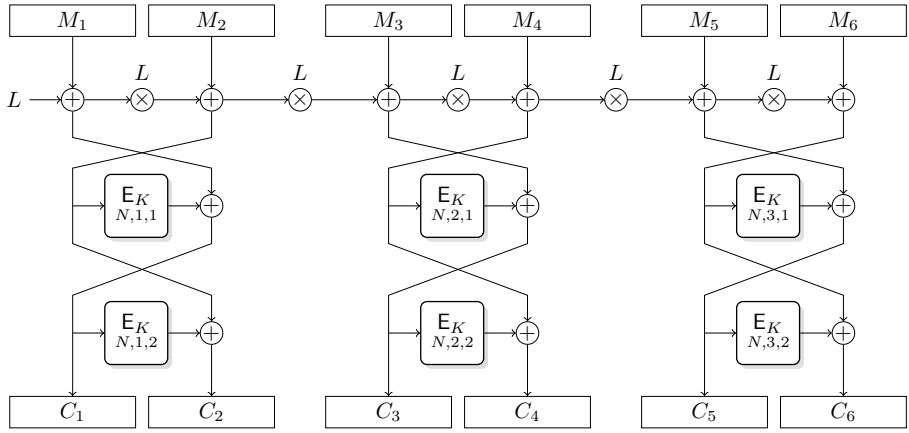


Figure 5.7: Processing plaintext. The value L is generated using the output of a block cipher call tweaked by the nonce.

5.6 Pseudorandom Functions

A useful inverse-less counterpart to the block cipher is the *pseudorandom function* (PRF), $\mathsf{F} : \mathsf{K} \times \mathsf{X} \to \mathsf{Y}$, which maps keys and elements of X into elements of Y; we assume that the sets K, X, and Y are finite. The quality of a PRF when keyed with a secret, uniformly generated value is measured by comparing it with a URF from X to Y.

Definition 5.6.1 (PRF)**.** Let $\mathsf{F} : \mathsf{K} \times \mathsf{X} \to \mathsf{Y}$ be a PRF. Then the pseudorandom function (PRF) advantage of adversary $\mathbf{A}$ against F is

$$\mathsf{PRF}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{F}_K \,;\, \pi)\,, \tag{5.50}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a URF from X to Y.

Note that we follow convention by using the term "PRF" to describe both a quality measure and a functionality. Hence a PRF is a function designed to have good PRF quality, and even though a block cipher is not designed to be a PRF, it could be used as one, and its quality as a PRF can be measured.

Proposition 5.6.1. *Let* $\mathsf{E} : \mathsf{K} \times \mathsf{X} \to \mathsf{X}$ *be a block cipher, then for any* PRF*-adversary* $\mathbf{A}$ *making at most* q *queries,*

$$\mathsf{PRF}(\mathbf{A}) \leq \frac{q(q-1)}{2\,|\mathsf{X}|} + \mathsf{PRP}(\mathbf{A})\,. \tag{5.51}$$

The proposition follows from an application of Lemma 2. Likewise, one could measure the PRP quality of a PRF and get the same bound, but measuring the SPRP quality of a PRF is meaningless since the PRF does not provide any inverse operation.

Example 5.6.2. A simple way of constructing a universal hash function using a PRF is by XORing outputs together, as described by Bellare, Guérin, and Rogaway [26]. Say that $\mathsf{X} = \{0,1\}^{n+s}$ and $\mathsf{Y} = \{0,1\}^m$, and let F be a URF from X to Y; in the actual construction the URF is replaced by a PRF. Then a message $(M_1, M_2, \ldots, M_\ell) \in \mathsf{X}^*$ is mapped to the value

$$\mathsf{F}_K(1_s M_1) \oplus \mathsf{F}_K(2_s M_2) \oplus \cdots \oplus \mathsf{F}_K(\ell_s M_\ell)\,. \tag{5.52}$$

Two messages $M = (M_1, M_2, \ldots, M_\ell)$ and $M' = (M'_1, M'_2, \ldots, M'_\ell)$ collide only if

$$\Big(\mathsf{F}_K(1_s M_1) \oplus \mathsf{F}_K(1_s M'_1)\Big) \oplus \cdots \oplus \Big(\mathsf{F}_K(\ell_s M_\ell) \oplus \mathsf{F}_K(\ell_s M'_\ell)\Big) = 0\,. \tag{5.53}$$

Since $M \neq M'$ there exists an i such that $M_i \neq M'_i$, hence the above equation contains a term of the form $\mathsf{F}_K(i_s M_i) \oplus \mathsf{F}_K(i_s M'_i)$ which is uniformly distributed, and independent of all other values, meaning the above equation will equal 0 with probability at most $1/|\mathsf{Y}|$. ◀

Often block ciphers are also used to construct PRFs, either directly, or by truncating the block cipher output, or by XORing two independently keyed block ciphers [30, 89, 118]. Conversely, PRFs can be used to construct block ciphers, via use of multiple rounds of a Feistel network [117].

PRFs also have a counterpart which explicitly allows for variable input lengths, VIL-PRFs. VIL-PRFs compress input just like universal hash functions, but also provide functionality beyond collision resistance. Their quality measure is identical to those of PRFs, except their ideal counterpart is extended to a family of URFs indexed by message length.

Definition 5.6.3 (VIL-PRF)**.** A *variable-input-length* URF $\pi : \mathsf{X}^* \to \mathsf{Y}$ is a family of URFs $\{\pi_i\}_{i \geq 0}$ where $\pi_i : \mathsf{X}^i \to \mathsf{Y}$ and $\pi(X) = \pi_{|X|}(X)$ for $X \in \mathsf{X}^*$.

Definition 5.6.4 (VIL-PRF Advantage)**.** Let $\mathsf{F} : \mathsf{K} \times \mathsf{M} \to \mathsf{X}$ be a VIL-PRF. Then the variable-input-length pseudorandom function (VIL-PRF) advantage of adversary $\mathbf{A}$ against F is

$$\mathsf{VIL\text{-}PRF}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{F}_K\,;\,\pi)\,, \tag{5.54}$$

where $K \xleftarrow{\$} \mathsf{K}$ and π is a VIL-URF.

Example 5.6.5. Besides its use as an encryption scheme, CBC mode has also been used as a VIL-PRF, by fixing the random IV input to 0, suppressing intermediate output, and only using the last ciphertext block as output. The resulting mode is referred to as CBC-MAC: given a message $M_1 M_2 \cdots M_\ell \in \mathsf{X}^*$, it computes

$$V_0 = 0^n \tag{5.55}$$

$$V_i = \mathsf{E}_K(M_i \oplus V_{i-1}) \quad \text{for } i = 1, \ldots, \ell\,, \tag{5.56}$$

and outputs V_ℓ. Yet, CBC-MAC is not secure as a PRF, which can be seen with the following attack: query $M \in \mathsf{X}$ and receive V, then query $V \in \mathsf{X}$ to receive V', finally check to see if the output resulting from query $(M, 0^n)$ is V'. This property is true for CBC-MAC, but not for a VIL-URF. Nevertheless, CBC-MAC works as a VIL-PRF if none of the messages share any prefixes [143], or all messages are of equal length [27]. ◀

A common way of creating a VIL-PRF is using *hash-then-encrypt*, which composes a universal hash function $\mathsf{F} : \mathsf{K} \times \mathsf{M} \to \mathsf{X}$ with either a PRF $\mathsf{E} : \mathsf{K}' \times \mathsf{X} \to \mathsf{Y}$, to form $\mathsf{E}_{K_1} \circ \mathsf{F}_{K_2}$, where $K_1 \xleftarrow{\$} \mathsf{K}'$ and $K_2 \xleftarrow{\$} \mathsf{K}$ are independent. Distinguishing the composition from a VIL-PRF amounts to either distinguishing E from a URF, and if E is indistinguishable from a URF, then finding a collision in F, which results in finding a collision for $\mathsf{E} \circ \mathsf{F}$.

Proposition 5.6.2. *Let* $\mathsf{F} : \mathsf{K} \times \mathsf{M} \to \mathsf{X}$ *be a universal hash function and* $\mathsf{E} : \mathsf{K}' \times \mathsf{X} \to \mathsf{Y}$ *a PRF, and let* $\mathbf{A}$ *be a VIL-PRF adversary against* $\mathsf{E} \circ \mathsf{F}$*, then*

$$\mathsf{VIL\text{-}PRF}(\mathbf{A}) \leq \mathsf{CB}_\mathsf{F} + \mathsf{PRF}_\mathsf{E}(\mathbf{A}(\circ\mathsf{F}))\,. \tag{5.57}$$

Subsequent application of Proposition 5.6.1 gives the bound for when E is a block cipher.

Example 5.6.6. A way of fixing CBC-MAC is by using it in the hash-then-encrypt construction; when composed with a block cipher the resulting VIL-PRF is sometimes called EMAC [36]. ◀

Example 5.6.7. One can use the PRF XOR universal hash from Example 5.6.2 to construct a parallelizable VIL-PRF, however the resulting construction would use two independent keys. An alternative is to use the same key for all PRF calls, but then to use a different counter value for the final PRF call instead of an independent key, as is done for the *protected counter sum* [37]. If the PRF is replaced with a tweakable block cipher, then the resulting construction corresponds to the VIL-PRF used to process tweaks in COPE, also known as PMAC [153]; see Figure 5.6a. ◀

Chapter 6

Constructions

In this chapter we discuss constructions which are able to achieve integrity, confidentiality, or both. The tools used to create them were introduced in Chapter 5.

We start the chapter by discussing methods of estimating the efficiency of the schemes, which will be necessary to discuss why certain schemes are more efficient than others. We see how choosing stronger security requirements decreases efficiency. Furthermore, we discuss how the efficiency with which integrity can be added to an encryption scheme to form an AE scheme depends on the encryption scheme's security, which in turn affects its efficiency as well. In Section 6.3.3 we discuss how to avoid the issue of ciphertext expansion with COPE, and in Section 6.4.3 we explain how to efficiently add integrity to COPE in order to form COPA.

6.1 Efficiency Heuristics

The only way to know a scheme's efficiency is to implement and test it. Nevertheless, understanding efficiency at a heuristic level gives designers goals to achieve. Focusing on modes of operation simplifies the measurements, since there are few objects that need to be taken into account. At the level of abstraction of a mode there are three useful measures: the number and types of operations, the parallelizability of the operations, and the state size.

Operations. The most commonly used operations in modes are XORs, finite field arithmetic, and calls to the underlying primitive, such as a block cipher. Out of these, the heaviest are the primitive calls and finite field multiplication. Measuring the number of heavy operations per unit plaintext, also known as the *rate*, can give an indication of how efficient the resulting scheme will be, relative to the efficiency of the heavy operations.

Finite field multiplication and primitive calls are treated as being equally expensive, since in practice, either could be more expensive than the other. For example, the operations differ in efficiency on different generations of Intel CPUs. On Nehalem and Sandy Bridge, finite field multiplication over $\mathrm{GF}(2^{128})$ runs slower than AES [111], whereas on Haswell, the opposite is true [87], when using the AES instruction sets.

Finally, the number of different operations used by a scheme can give an indication as to how large its implementation will be in hardware. For example, a scheme using both a primitive and its inverse will most likely be larger in hardware implementation size than a scheme not using the inverse primitive.

Parallelizability. A scheme is parallelizable if it can perform many of its primitive calls and multiplications independently. Some schemes have a certain amount of operations which must be performed serially; for example, when the input to one block cipher call is the output of another as in TC3 (Example 5.4.4). If a significant amount of these operations must be performed serially, then we do not call the scheme parallelizable.

Parallelizability can lead to a significant increase in efficiency. If the underlying primitive is AES, then the AES-NI instruction set on Intel and AMD CPUs enables significant parallelization, sometimes allowing for an improvement of a factor three or more; see, for instance, the difference between CBC encryption and decryption [5].

State Size. The state size of a scheme is the maximum amount of data that an algorithm would need to keep in memory as it is processing messages. In the worst case, schemes would need to keep data which is at least as large as the input. In the best case, schemes are able to process the input using a constant state size, assuming they may output data as they receive it; such schemes are called *online*. Note that in conventional AE, it is difficult for the decryption algorithm to be online, since decrypted plaintext should not be released until verification is complete in order to ensure security.

6.2 MAC Algorithms

Message Authentication Code (MAC) algorithms are authenticators which output the message in the clear, and generate a tag with which integrity is checked. The Tag_K algorithm of a MAC uses some function ρ to compress the message into a tag T, and outputs both the message and tag:

$$\mathsf{Tag}_K(M) \stackrel{\text{def}}{=} (M, \rho_K(M))\,. \tag{6.1}$$

The verification algorithm receives a message-tag pair, and checks validity of the pair by using the message and key to regenerate a tag with ρ, and compares ρ's output with the given tag:

$$\mathsf{Ver}_K(M,T) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \rho_K(M) = T \\ 0 & \text{otherwise}\,. \end{cases} \tag{6.2}$$

In this case $\mathsf{S} = \{1\}$ and $\mathsf{F} = \{0\}$.

6.2.1 Nonce IV

The Wegman-Carter construction for MACs [174] uses a universal hash function to compress long messages into a short output, which is then XORed with the output of a primitive call, such as a block cipher or PRF. The primitive uses a different key than the universal hash function, and the primitive's input is a nonce; see Figure 6.1 for a diagram.

If the primitive is a PRF, then the outputs of the Wegman-Carter construction are independent and uniformly distributed (Lemma 1). In particular, constructing a forgery without using the PRF's output will result in low forgery probability. Consider for example the forgery attempt (N', M', T'), then $\mathsf{Ver}_K(N', M') = 1$ only if

$$\mathsf{UH}_{K_1}(M') \oplus \pi_{K_2}(N') = T'\,. \tag{6.3}$$

If N' has never been queried to π before, then π's output is independent and uniformly distributed, meaning the above equation will be satisfied with low probability. If $N' = N$ for some previous query (N, M) with output T, then $\mathsf{Ver}_K(N', M') = 1$ only if

$$\mathsf{UH}_{K_1}(M') \oplus \mathsf{UH}_{K_1}(M) = T' \oplus T\,, \tag{6.4}$$

As as long as $M \neq M'$, this is exactly the additive collision bound of the universal hash function. If $M = M'$, then T cannot equal T', which means that the forgery fails anyway.

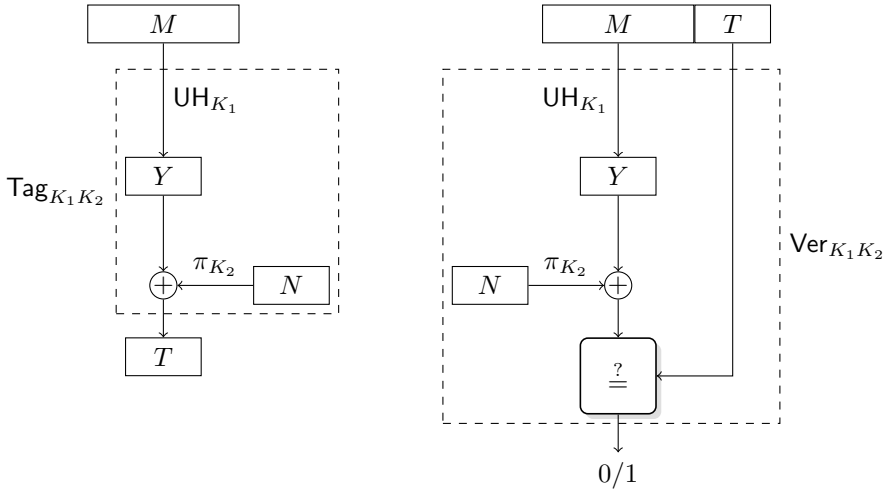


Figure 6.1: A Wegman-Carter construction with universal hash UH and primitive π. The tagging algorithm is on the left and the verification algorithm on the right.

The above argument holds for PRPs or block ciphers as well. The following theorem by Bernstein [38, Theorem 5.1] reduces the security of Wegman-Carter MACs to the additive collision bound of the universal hash, and the so-called maximum q *interpolation probability* of the primitive, which is

$$\max_{X \in \mathsf{X}^q, Y \in \mathsf{Y}^q} \mathbf{P}\left[\pi_{K_2}(X_1) = Y_1, \pi_{K_2}(X_2) = Y_2, \ldots, \pi_{K_2}(X_q) = Y_q\right] . \tag{6.5}$$

Theorem 7 ([38]). *Consider a Wegman-Carter construction with universal hash* $\mathsf{UH} : \mathsf{M} \to \mathsf{Y}$ *and primitive* $\pi : \mathsf{IV} \to \mathsf{Y}$. *Say that* $|\mathsf{IV}| \leq |\mathsf{Y}|$ *and that* π *has maximum* q*-interpolation probability at most* $\delta / |\mathsf{Y}|^q$ *and maximum* $(q+1)$*-interpolation probability at most* $\delta \cdot \mathsf{ACB}_\rho / |\mathsf{Y}|^q$. *Let* $\mathbf{A}$ *be an* $\mathsf{n\text{-}Int}$*-adversary making at most* q *tagging queries and* v *verification queries, then*

$$\mathsf{n\text{-}Int}(\mathbf{A}) \leq v \cdot \delta \cdot \mathsf{ACB}_\rho . \tag{6.6}$$

Many Wegman-Carter constructions use polynomial-based universal hash functions. The XOR MAC [26] can be viewed as a type of Wegman-Carter construction using a PRF: it uses the XOR universal hash construction from Example 5.6.2, but instead of keying the primitive π_{K_2} with an independent key, it uses the same PRF from the universal hash, but with a different counter, allowing one to use Bernstein's theorem.

Repeating the IV could result in an attack against Wegman-Carter constructions, as described by Handschuh and Preneel [90] and Joux [103].

6.2.2 Deterministic MACs

As explained in Chapter 3, achieving Int-security in the abused IV setting is possible. In fact, many MAC algorithms are deterministic and do not require an IV input. The advantage that such deterministic MACs have over nonce IV MACs is that there is no IV to send, thereby reducing communication costs. However, dropping the IV comes at the cost of a slight loss in security, as explained in Chapter 8.

Deterministic MACs usually use VIL-PRFs as their basic building block, meaning the function ρ in Equations (6.1) and (6.2) is a VIL-PRF. The best bound for the hash-then-encrypt constructions from Section 5.6 using a PRP as primitive was published by Dodis and Pietrzak [70, Proposition 1].

Theorem 8 ([70])**.** *Consider the hash-then-encrypt construction with the primitive a PRP and universal hash function* $\mathsf{F} : \mathsf{M} \to \mathsf{Y}$*. Let* $\mathbf{A}$ *be an* Int*-adversary making no more than* q *tagging queries and* v *verification queries. If* $\mathsf{CB}_{\mathsf{F}} \geq 1/(|\mathsf{Y}| - q)$*, then*

$$\mathsf{Int}(\mathbf{A}) \leq \mathsf{CB}_{\mathsf{F}} \cdot (q^2 + v) \ . \tag{6.7}$$

In terms of efficiency, deterministic and nonce IV MACs are roughly equivalent. Generally, MACs only use one heavy operation per plaintext block, although more might be needed if better security is required; see Chapter 8. Although popular deterministic MACs, such as ECBC, are serial, it is possible to construct parallelizable ones, such as PMAC. Many nonce IV MACs are parallelizable, including the polynomial-based ones.

6.3 Encryption Schemes

6.3.1 Nonce and Random IV

Some of the conceptually simplest modes are those which only provide CPA confidentiality in the nonce and random IV settings. Chapter 5 contains three examples, namely CTR mode, CBC mode, and simplified OCB encryption. Out of the three, CBC mode has the least overhead, using only XOR in addition to the block cipher calls. CTR mode requires an additional counter to be

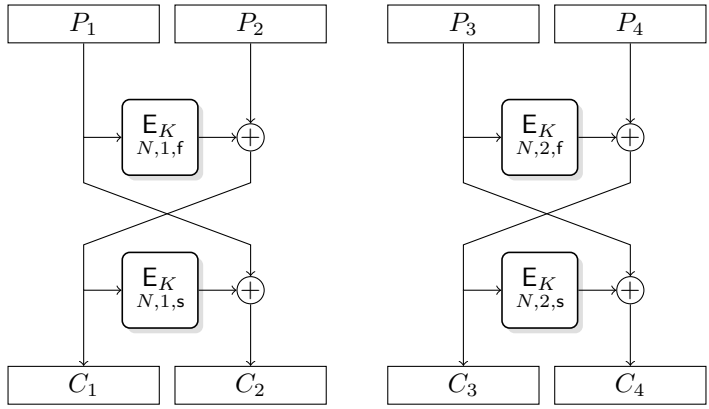


Figure 6.2: OTR encryption on four blocks of plaintext.

generated, but has the advantage of being completely parallelizable in both encryption and decryption. Furthermore, CTR mode only uses forward calls to the block cipher, allowing its implementation size in hardware to be smaller and reducing the block cipher quality requirements.

Like CTR mode, OCB encryption is parallelizable in both encryption and decryption, yet it adds extra overhead via its use of tweakable block ciphers, which are implemented using the XEX construction (Example 5.2.4). Furthermore, like CBC, OCB requires the use of both forward and inverse block cipher calls. Hence OCB encryption does not seem to improve upon CTR mode encryption, yet later we will see that adding integrity to OCB encryption can be done much more efficiently than with CTR mode.

The encryption of OTR mode [127] removes use of the inverse block cipher call by using a Feistel network, as depicted in Figure 6.2, while still maintaining parallelizability of the block cipher calls. However, in comparison with OCB, the parallelizability is reduced since two blocks must be processed sequentially. Like OCB, it adds overhead over CTR mode, but, again, it is much simpler to add integrity.

None of the above schemes achieve CCA-security. For the nonce-based schemes, simply pick an IV N and ciphertext C, decrypt it, and then encrypt it: when interacting with $(\mathsf{Enc}_K, \mathsf{Dec}_K)$ you get $\mathsf{Enc}_K^N(\mathsf{Dec}_K(N, C)) = C$, whereas when interacting with $(\mathsf{Enc}_K \circ \$, \mathsf{Dec}_K)$, you get

$$\mathsf{Enc}_K^N(\$(\mathsf{Dec}_K(N, C)))\,, \tag{6.8}$$

which equals C with low probability.

For CBC, the same attack cannot be applied because Enc always receives an independent, uniformly generated IV, and in fact, in the random IV setting the attack would not work for CTR, OCB, and OTR encryption. However, a different attack applies to CBC. One can pick a two-block plaintext P_1P_2, and encrypt it to receive C_1C_2, and the IV R. Then, one decrypts the ciphertext C_1 with IV R and C_2 missing. Since C_1 does not equal C_1C_2, it is a valid ciphertext, which decrypts to P_1. When interacting with CBC one always receives P_1, whereas when interacting with $\mathsf{Enc}_K \circ \$$, one does not receive P_1 with high probability. Even in the random IV setting similar attacks will apply to CTR, OCB encryption, and OTR encryption.

A straightforward way of achieving CCA-security is by using a tweakable cipher. One might try to convert the tweakable cipher E into an encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ by tweaking the cipher with a nonce, and encrypting the plaintext using the given permutation:

$$\mathsf{Enc}_K^N(P) \stackrel{\text{def}}{=} (N, \mathsf{E}_K^N(P)) \tag{6.9}$$

$$\mathsf{Dec}_K(N, C) \stackrel{\text{def}}{=} \mathsf{D}_K^N(C)\,. \tag{6.10}$$

However the construction $(\mathsf{Enc}, \mathsf{Dec})$ does not achieve CCA-security for the same reason that the CTR mode does not achieve CCA-security: decrypting and encrypting should result in the same ciphertext, which it does not when interacting with $\mathsf{Enc} \circ \$$. The issue is that every ciphertext will decrypt to some plaintext with the known property that encryption of that plaintext should result in the original ciphertext.

A simple way of breaking this property is by adding redundancy, commonly known as encode-then-encipher [34]. The redundancy can be as simple as including a constant block of plaintext $P_0 \in \mathsf{X}^n$:

$$\mathsf{Enc}_K^N(P) \stackrel{\text{def}}{=} (N, \mathsf{E}_K^N(P, P_0)) \tag{6.11}$$

$$\mathsf{Dec}_K(N, C) \stackrel{\text{def}}{=} \mathsf{D}_K^N(C)\,. \tag{6.12}$$

Using a SPRP cipher, one achieves n-CCA-security since adversaries would have to find a ciphertext C such that $\mathsf{D}_K^N(C)$ is of the form (P, P_0). The reduction from n-CCA-adversary $\mathbf{A}$ to SPRP adversary $\mathbf{B}$ simply consists of converting $\mathbf{A}$'s queries to $(\mathsf{Enc}, \mathsf{Dec})$ to E-queries via Equations (6.11) and (6.12). Define $\mathbf{B}'$ to be $\mathbf{B}$ which also prepends $\$^{(\cdot)}$ to any Enc query.

Theorem 9. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be the construction defined above using tweakable cipher* E. *Then for any* n-CCA*-adversary* $\mathbf{A}$ *against* $(\mathsf{Enc}, \mathsf{Dec})$ *making at most*

d queries to Dec*, we have*

$$\mathsf{n\text{-}CCA}_{(\mathsf{Enc},\mathsf{Dec})}(\mathbf{A}) \leq \mathsf{SPRP}_{\mathsf{E}}(\mathbf{B}) + \mathsf{SPRP}_{\mathsf{E}}(\mathbf{B}') + \frac{2d}{|X|^n - d}\,. \tag{6.13}$$

Proof. Let $(\mathsf{Enc}[\pi], \mathsf{Dec}[\pi])$ denote $(\mathsf{Enc}, \mathsf{Dec})$ with (E, D) replaced by the tweakable URP (π, π^{-1}). Using the triangle inequality we get

$$\mathsf{n\text{-}CCA}_{\mathsf{Enc},\mathsf{Dec}}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}, \mathsf{Dec}_K\,;\, \mathsf{Enc}_K^{(\cdot)} \circ \$^{(\cdot)}, \mathsf{Dec}_K) \tag{6.14}$$

$$\leq \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}, \mathsf{Dec}_K\,;\, \mathsf{Enc}_K^{(\cdot)}[\pi], \mathsf{Dec}_K[\pi]) \tag{6.15}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}[\pi], \mathsf{Dec}_K[\pi]\,;\, \mathsf{Enc}_K^{(\cdot)}[\pi] \circ \$^{(\cdot)}, \mathsf{Dec}_K[\pi]) \tag{6.16}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}[\pi] \circ \$^{(\cdot)}, \mathsf{Dec}_K[\pi]\,;\, \mathsf{Enc}_K^{(\cdot)} \circ \$^{(\cdot)}, \mathsf{Dec}_K)\,. \tag{6.17}$$

The first term is bounded by $\mathsf{SPRP}(\mathbf{B})$ and the third by $\mathsf{SPRP}(\mathbf{B}')$. We focus on the second term.

The only way $\mathbf{A}$ will successfully distinguish is by making $\mathcal{O}_1$ queries, since $\mathcal{O}_2$ is the same on both sides. Consider a query $\mathcal{O}_1^N(P)$. When interacting with both $\mathsf{Enc}[\pi]$ and $\mathsf{Enc}[\pi] \circ \$$, the query gets converted to $\pi^N(P', P_0)$, where P' is either P or $\$(P)$. In the latter case, since N is never repeated for any encryption queries, $\pi^N(P', P_0)$ will always output independent, uniformly distributed values, unless $(P', P_0) = \pi^{-1N}(C)$ for some query $\mathsf{Dec}_K(N, C)$. Finding a $\mathsf{Dec}_K(N, C)$ query which contains P_0 in its right half can be done with probability at most $d/(|\mathsf{X}|^{|P_0|} - d)$ by fixing N and query different ciphertexts.

Similarly, since N is never repeated, $\pi^N(P, P_0)$ always outputs independent, uniformly distributed values, unless $(P, P_0) = \pi^{-1N}(C)$ for some query $\mathsf{Dec}_K(N, C)$, which occurs with probability at most $1/(|\mathsf{X}|^{|P_0|} - d)$. Therefore the distinguishing advantage of any adversary making at most d decryption queries, is bounded above by

$$\frac{2d}{|X|^n - d}\,. \tag{6.18}$$

□

6.3.2 Abused IV

Neither CTR, CBC, OCB, nor OTR encryption modes achieve CPA security in the abused-IV setting. Repeating the IV in CTR mode means receiving

$S \oplus P_1$ and $S \oplus P_2$ as ciphertexts, where S is the stream of block cipher outputs. By XORing together the ciphertexts one gets $P_1 \oplus P_2$ which is a breach of confidentiality. Repeating the IV in OCB encryption means that repeated blocks of plaintext will show up as repeated blocks of ciphertext, another breach of confidentiality. Similar attacks can be applied to CBC and OTR, and Fleischmann et al. [77] discuss others.

The tweakable cipher construction discussed in Equations (6.11) and (6.12) actually achieves $\mathsf{a\text{-}CCA}$ security: repeating the IV results in picking the same permutation, and doing so leaks repetition of the plaintext, and nothing else.

Theorem 10. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be the construction defined in Equations* (6.11) *and* (6.12) *with tweakable cipher* E *over plaintexts* X^**. Then for any* $\mathsf{a\text{-}CCA}$*-adversary* $\mathbf{A}$ *against* $(\mathsf{Enc}, \mathsf{Dec})$ *making at most* q *encryption queries of length at least* ℓ *and at most* d *decryption queries,*

$$\mathsf{a\text{-}CCA}_{\mathsf{Enc},\mathsf{Dec}}(\mathbf{A}) \leq \mathsf{SPRP}_{\mathsf{E}}(\mathbf{B}) + \mathsf{SPRP}_{\mathsf{E}}(\mathbf{B}') + \frac{q^2}{|\mathsf{X}|^{\ell}} + \frac{2d}{|\mathsf{X}|^n - d}\,, \tag{6.19}$$

where $\mathbf{B}$ *and* $\mathbf{B}'$ *are the same reductions as from Theorem 9.*

Proof. The first part of the proof is identical to the proof of Theorem 9, hence we focus on

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}[\pi], \mathsf{Dec}_K[\pi]\,;\, \mathsf{Enc}_K^{(\cdot)}[\pi] \circ \$^{(\cdot)}, \mathsf{Dec}_K[\pi])\,. \tag{6.20}$$

In contrast with the proof of Theorem 9, the IV is no longer unique for every encryption. In particular, $\mathcal{O}_K^N(P)$ is independent of $\mathcal{O}_K^{N'}(P')$ for every (N', P') with $N \neq N'$, but if N is repeated, then we know that

$$\mathsf{Enc}_K^N(P) = (N, \mathsf{E}_K^N(P, P_0)) \neq (N, \mathsf{E}_K^N(P', P_0)) = \mathsf{Enc}_K^N(P') \tag{6.21}$$

for $P \neq P'$, whereas with $\mathsf{Enc}_K \circ \$$ this could occur with probability $1/\,|\mathsf{X}|^{|P|}$ if $|P| = |P'|$. Since the distribution of Enc_K^N is identical to the distribution of $\mathsf{Enc}_K^N \circ \$^N$, as long as $\$(P) \neq \$(P')$ for two different queries $P \neq P'$, and $(P, P_0) \neq \mathsf{Dec}_K(N, C)$, we have that the advantage of any adversary is bounded above by the advantage of causing either of those two events, which is at most

$$\frac{q^2}{|\mathsf{X}|^{\ell}} + \frac{2d}{|\mathsf{X}|^n - d}\,. \tag{6.22}$$

□

The downside to using tweakable ciphers when implemented as modes of operation, is that they usually require several calls to the underlying block

cipher per plaintext block. Furthermore, they require multiple passes over the plaintext, which means a sufficiently large state is needed in order to store data which is roughly as long as the plaintext.

At the cost of achieving the comparatively weaker oa-CPA security, tweakable online ciphers provide an efficient alternative to using tweakable ciphers. By incorporating a nonce into the tweak, a tweakable online cipher $\mathsf{E} : \mathsf{K} \times \mathsf{P} \to \mathsf{C}$ can be converted into an encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ via

$$\mathsf{Enc}_K^N(P) \stackrel{\text{def}}{=} (N, \mathsf{E}_K^N(P)) \tag{6.23}$$

$$\mathsf{Dec}_K(N, C) \stackrel{\text{def}}{=} \mathsf{D}_K^N(C)\,. \tag{6.24}$$

Theorem 11. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *denote the encryption scheme constructed from the tweakable online cipher* E *over* X^**, then for any adversary* $\mathbf{A}$*,*

$$\mathsf{oa\text{-}CPA}_{\mathsf{Enc}}(\mathbf{A}) \leq \mathsf{o\text{-}PRP}_{\mathsf{E}}(\mathbf{A}) + \frac{q^2}{|\mathsf{X}|}\,. \tag{6.25}$$

Proof.

$$\mathsf{oa\text{-}CPA}_{\mathsf{Enc}}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Enc}_K^{(\cdot)}\,;\, \mathsf{Enc}_K^{(\cdot)} \circ \$^{(\cdot)}) \tag{6.26}$$

$$= \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K^{(\cdot)}\,;\, \mathsf{E}_K^{(\cdot)} \circ \$^{(\cdot)})) \tag{6.27}$$

$$\leq \underset{\mathbf{A}}{\Delta}(\mathsf{E}_K^{(\cdot)}\,;\, \pi^{(\cdot)}) + \underset{\mathbf{A}}{\Delta}(\pi^{(\cdot)}\,;\, \pi^{(\cdot)} \circ \$^{(\cdot)}(\cdot)) \tag{6.28}$$

$$= \mathsf{o\text{-}PRP}_{\mathsf{E}}(\mathbf{A}) + \underset{\mathbf{A}}{\Delta}(\pi^{(\cdot)}\,;\, \pi^{(\cdot)} \circ \$^{(\cdot)})\,. \tag{6.29}$$

Since $\pi^{(\cdot)}$ is indistinguishable from $\pi^{(\cdot)} \circ \$^{(\cdot)}$ with loss $q^2/|\mathsf{X}|$, we have our result. □

6.3.3 Avoiding Ciphertext Expansion

Encryption schemes which map plaintexts to ciphertexts of the same length are desirable, since they do not increase the amount of data that needs to be communicated. So far CTR mode is the only encryption scheme which was presented as being length-preserving. Avoiding so-called *ciphertext expansion* is easy to do in CTR mode since the block cipher outputs just need to be

truncated to match the plaintext length. Preserving length in other modes is non-trivial.

Take CBC mode for example. In Chapter 5, Example 5.1.4, CBC mode was only presented as operating on plaintexts which were made of full blocks. So a four-block plaintext $P = P_1P_2P_3P_4$ is encrypted to four-block ciphertext $C = C_1C_2C_3C_4$ using random IV R via

$$C_0 = R \tag{6.30}$$

$$C_i = \mathsf{E}_K(C_{i-1} \oplus P_i)\,, \tag{6.31}$$

and decryption works via

$$C_0 = R \tag{6.32}$$

$$P_i = \mathsf{D}_K(C_i) \oplus C_{i-1}\,. \tag{6.33}$$

Since decryption works by calling the inverse block cipher on each ciphertext block, truncating ciphertext blocks is not possible without making decryption impossible. A trick used to get around this restriction is *ciphertext stealing* [64], which works as follows. If P_4 is not a complete block, then pad P_4 with zeros until it is, to create P_4'. Proceed by encrypting $P_1P_2P_3P_4'$, with resulting ciphertext $C_1C_2C_3C_4$. Truncate C_3 to be the same length as P_4, resulting in C_3', and send the ciphertext $C_1C_2C_3'C_4$. Then decryption works as usual for C_1 and C_2, and before C_3' is decrypted, $\mathsf{D}_K(C_4) = P_4' \oplus C_3$ is computed, which contains the missing part of C_3' necessary to complete the decryption. The encryption process is depicted in Figure 6.3. Rogaway, Wooding, and Zhang [157] provide a formal analysis of why ciphertext stealing preserves security.

Ciphertext stealing works for CBC mode because each ciphertext block can be processed independently of the others during decryption. Applying ciphertext stealing to the tweakable online cipher TC3 does not work since ciphertext block C_i is necessary in order to decrypt ciphertext block C_{i+1}. COPE's decryption on the other hand, only needs pairs of ciphertext blocks in order to decrypt a plaintext. Specifically, knowing just ciphertext blocks C_3 and C_4, one can determine plaintext P_4 via

$$P_4 = \mathsf{D}_K^{(4,3)}\Big(\mathsf{D}_K^{(3,4)}(C_3) \oplus \mathsf{D}_K^{(4,4)}(C_4)\Big)\,; \tag{6.34}$$

see also Figure 6.4. As a result, ciphertext can be "stolen" from C_2 in order to pad P_4. Since COPE must work when IVs are repeated, the last tweakable block cipher calls must be tweaked differently from the case when the last block is full. A similar trick can be applied to COBRA, although the process is slightly more involved; see Appendix A for a description.

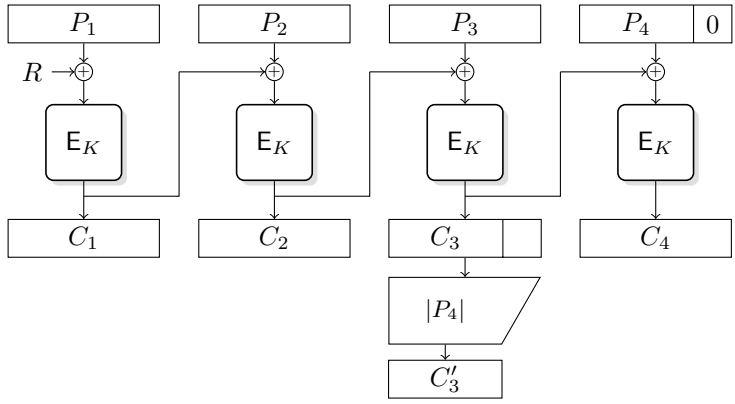


Figure 6.3: CBC mode with ciphertext stealing.

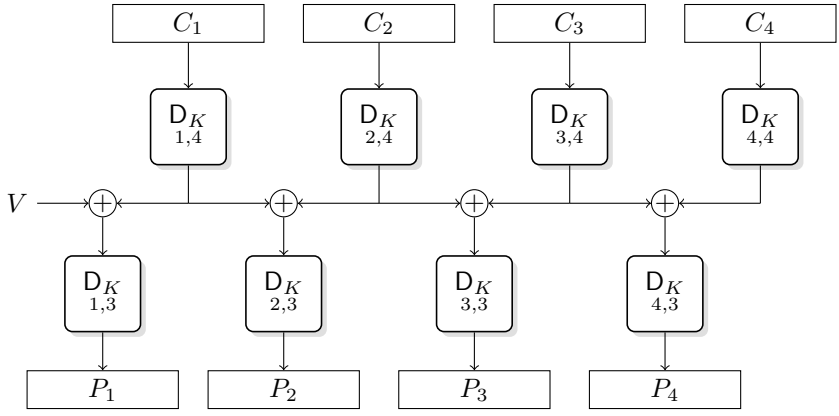


Figure 6.4: COPE decryption. The value V is computed as in Figure 5.6.

Nevertheless, ciphertext stealing for COPE only works with plaintexts which are at least two blocks long. An alternative is to use a tweakable cipher which works on $\{0,1\}^{\leq 3n}$, although the construction of such ciphers is non-trivial. One example is XLS [149], which, given a cipher that can process plaintexts of length l, can expand the input to plaintexts of length $l + s$ bits for any $s < n$. However, XLS was shown not to be SPRP by Nandi [135], resulting in an attack against COPA [137], an extension of COPE used to handle integrity (see Section 6.4.3). An alternative is THEM [182], which uses a combination of block cipher calls and finite field multiplications.

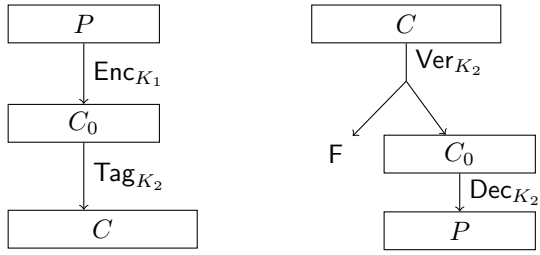


Figure 6.5: Encrypt-then-MAC.

6.4 AE Schemes

6.4.1 Generic Composition

One of the first methods developed to achieve AE is *generic composition* [32], which combines the use of a MAC together with an encryption scheme. Let $(\mathsf{Tag}, \mathsf{Ver})$ be a MAC and $(\mathsf{Enc}, \mathsf{Dec})$ an encryption scheme, then the Encrypt-then-MAC construction $(\mathsf{Aenc}, \mathsf{Adec})$ first encrypts plaintext using Enc_{K_1} under key K_1, then processes the resulting ciphertext using Tag_{K_2} with key K_2:

$$C_0 = \mathsf{Enc}_{K_1}(P) \tag{6.35}$$

$$C = \mathsf{Tag}_{K_2}(C_0)\,. \tag{6.36}$$

Decryption first checks whether $\mathsf{Ver}_{K_2}(C) \in \mathsf{F}$, and if it is not, then it outputs $\mathsf{Dec}_{K_2}(\mathsf{Ver}_K(C))$. Figure 6.5 displays a diagram of the process. As shown by Bellare and Namprempre [32], if K_1 and K_2 are independent, the MAC is Int-secure, and the encryption scheme CPA-secure, then the resulting AE scheme is Int-secure and CCA-secure.

Bellare and Namprempre [32] discuss several other natural constructions, and conclude that Encrypt-then-MAC is the only way to generically ensure that the resulting construction is secure. However, their approach uses the most general formalization of encryption schemes and authenticators. In contrast, Namprempre, Rogaway, and Shrimpton [130] explore what possible constructions there are when looking at random and nonce-IV based schemes, and discover many other ways of generically constructing secure AE schemes.

6.4.2 Dedicated Nonce-IV AE

The advantage to using generic composition is that it combines two constructions which are well-understood in order to achieve AE. Furthermore, on a theoretical level it establishes that there is nothing more to AE than composing a scheme that offers confidentiality with a scheme that offers integrity. Yet generic composition does not take advantage of any possible efficiency gains there might be from building an AE scheme using simpler components. Furthermore, it requires the use of two independent keys, whereas it might be possible to create AE schemes which only require one.

The *Galois Counter Mode*, or GCM, combines a polynomial-based Wegman-Carter MAC with CTR mode into a scheme which uses a single block cipher key. It can be viewed as an "encrypt-then-MAC" style AE scheme: first GCM encrypts the plaintext using CTR mode, and then it passes the ciphertext together with the associated data through the Wegman-Carter MAC.

Yet, other than the reduction in key size, GCM does not offer a big advantage over generic composition in terms of efficiency. Rather than adding a separate MAC, one could try to add integrity in a more efficient way to CTR mode. But doing so is not obvious. CTR mode is, in a sense, "too efficient", since there do not seem to be any extra values generated during the encryption process which could be used for an integrity check: the block cipher outputs are generated using the counter values which are independent of the plaintext, making them unsuitable, and all there is besides the block cipher outputs is the ciphertext, which would end up being an encrypt-then-MAC approach.

In contrast, OCB is able to add integrity by simply XORing together the plaintext blocks and passing it through a tweakable block cipher call. This is surprising since passing the XOR of message blocks through a tweakable block cipher call would not work as a MAC, since one could always swap message blocks to create a forgery, even in the nonce IV setting. The reason it works for OCB is because OCB's decryption algorithm will only output the right plaintext blocks if the right nonce is used and the ciphertext blocks are in their correct relative positions, otherwise one of the plaintext blocks will be the output of a arbitrary tweakable block cipher call, which means the resulting XOR will be unpredictable. OTR works similarly, and in fact only requires the XOR of half of the message blocks due to the use of the Feistel network.

6.4.3 Abused-IV AE

GCM and OCB fail to provide security when the IV is repeated, since confidentiality breaks down as pointed out in Section 6.3.2. Generic composition also only provides security if the underlying encryption scheme and MAC are secure in the abused IV setting.

One could compose a tweakable cipher as an encryption scheme with a deterministic MAC to get abused-IV security via the above result. Yet there is a more efficient way of adding integrity to a tweakable cipher, namely via the encode-then-encipher approach:

$$\mathsf{Aenc}_K^N(P) = (N, \mathsf{E}_K^N(P, P_0)) \tag{6.37}$$

$$\mathsf{Adec}_K(N, C) = \begin{cases} \mathsf{D}_K^N(C) & \text{if } \mathsf{D}_K^N(C) = (P, P_0) \\ \perp & \text{otherwise.} \end{cases} \tag{6.38}$$

Both Bellare and Rogaway [34] and Shrimpton and Terashima [165] analyze a more general version of the construction where the padding is replaced by an encoding function. Integrity is achieved since it is difficult to find a new ciphertext and nonce where decryption leads to the last plaintext block equaling P_0.

Achieving abused-IV AE with a tweakable cipher is straightforward, but not the most efficient method. Schemes such as SIV [156], BTM [99], and HBS [100] do so without using tweakable ciphers. As with tweakable ciphers, the downside to these schemes is that they require internal state large enough to fit data roughly the size of the plaintext.

Alternatively, one could attempt to add an efficient integrity check to encryption schemes built using online ciphers. Bellare et al. [24] give a few generic transformations to turn an online cipher into a secure authenticated encryption scheme, but their solutions require randomness. The McOE family [76] modifies Bellare et al.'s approach to efficiently add a deterministic integrity check to TC3. By appending the output of the IV encryption to the plaintext, an additional ciphertext block is produced, which can be viewed as a tag. If an adversary wants to create a forgery, then it must change an intermediate ciphertext block, which changes the tweaks used, and results in an unpredictable tag. The trick can be generalized to any online cipher that is o-SPRP secure, and can therefore be applied to POE as well, resulting in the construction POET [3].

However, the McOE trick only works with online ciphers that are o-SPRP secure, and does not work when attempting to add integrity to COPE since decryption of a plaintext block in COPE only depends on two ciphertext blocks

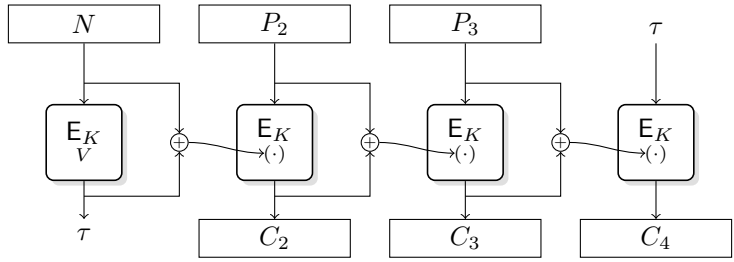


Figure 6.6: Add an integrity check to TC3.

(see Figure 6.4), and a change in the IV in decryption would not propagate to the end of the ciphertext processing.

Yet the OCB trick does work, namely computing the XOR of the plaintext and passing the result through extra block cipher calls. The tag T is computed by keeping an XOR checksum of the message blocks $\Sigma \stackrel{\text{def}}{=} M_1 \oplus \cdots \oplus M_\ell$ and computing

$$T \leftarrow \mathsf{E}_K^{(\ell,6)}\big(\mathsf{E}_K^{(\ell,5)}(\Sigma) \oplus S\big),$$

with $S \stackrel{\text{def}}{=} V_\ell$ denoting the last intermediate value in COPE's block chaining, as in Figure 5.6. The tweaks $(\cdot, 5)$ and $(\cdot, 6)$ are used to distinguish tag computation from encryption; see Figure 6.7. Tag verification occurs by checking if

$$\mathsf{E}_K^{(\ell,6)}\left(S \oplus \mathsf{E}_K^{(\ell,5)}(\Sigma)\right) = T,$$

where the tag is rejected if the equality is not true. The resulting scheme is called COPA [13].

One might conjecture that the OCB trick works for any o-PRP, yet it actually relies on the fact that block ciphers "destroy" relationships among plaintext blocks. Consider applying an OCB-type trick to the COBRA cipher, namely using an integrity check similar to OTR and ManTiCore [9]; see Figure 6.8. The trick works in the nonce IV setting, but once IVs can be abused, relationships among decrypted plaintext can be created to construct a forgery, as shown by Nandi [132–134]. Part of the reason why this attack works for COBRA and not for COPE is because COBRA uses finite field multiplication to create dependency upon preceding plaintext blocks in encryption, whereas COPE uses block cipher calls.

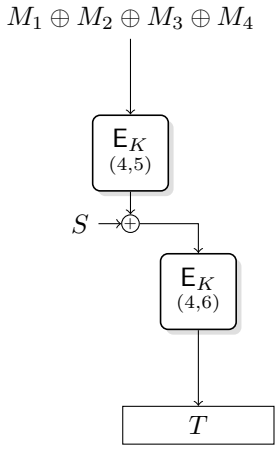


Figure 6.7: Adding an integrity check to COPE. The resulting scheme is called COPA.

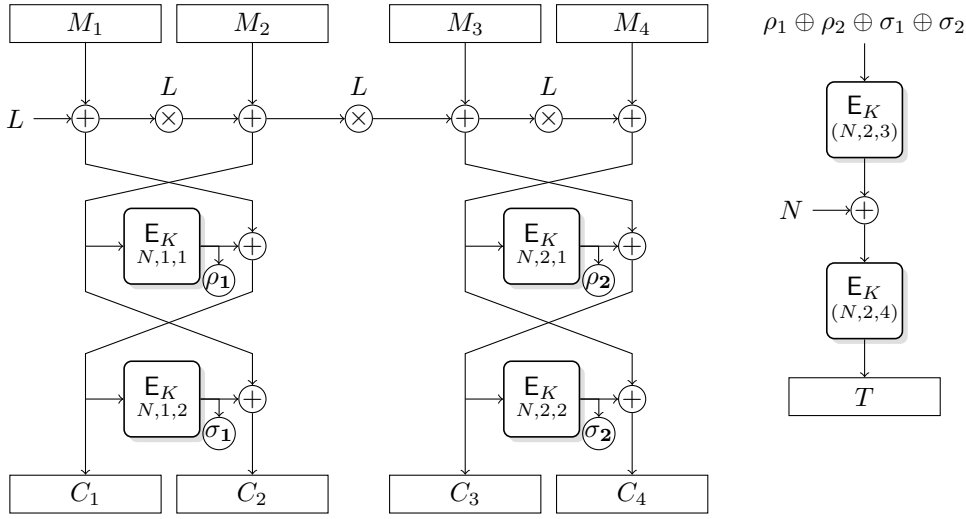


Figure 6.8: Computing the tag in COBRA. The outputs of the block cipher calls, ρ_i and σ_i, are XORed together and passed through two additional block cipher calls with different tweaks.

Chapter 7

Breaking Basic Security Assumptions

As seen in Chapter 3, security definitions might not initially reflect the actual environments in which schemes are used. The IV and online encryption extensions allow one to understand the worst- and best-case scenarios in less-than-ideal environments. For AE in particular, implementations in practice occur in environments which deviate slightly from those considered in the conventional security definitions, resulting in a violation of integrity, CCA, or even CPA security.

For example, the definitions in Chapter 3 assume that ciphertexts are output in one piece, whereas on-the-fly ciphertext output, where ciphertext fragments are output as plaintext is received, is common in practice. For example, SSH BPP processes fragmented ciphertexts which enables an attack recovering the first 32 bits of plaintext using only ciphertext [7], despite having undergone formal security analysis [29]. Extensions of the standard security definition to model these scenarios includes the so-called *blockwise adaptive* definitions, where chosen plaintext attacks surface [104] and combining CPA security with integrity no longer guarantees CCA security [78], and the formalization of Boldyreva et al. [50,53], where they also deal with boundary hiding and fragmentation-enabled denial-of-service attacks.

Besides omitting fragmented ciphertexts, the definition of AE from Chapter 3 also assumes that faulty verification must result in a single error message, and that plaintext coming from decryption can only be output upon successful verification. Yet, deviations from both of these requirements occur as well.

By outputting multiple error messages, adversaries can determine plaintext properties, which happens, for example, in Vaudenay's padding oracle attacks [170], where error messages or lack of acknowledgment indicate whether the unverified plaintext is correctly padded. Canvel et al. [58] show how to mount a padding oracle attack on the then-current version of OpenSSL by exploiting timing differences in the decryption processing of TLS. As shown by Paterson and AlFardan [8, 142] for TLS and DTLS, it is difficult to prevent attackers from learning decryption failure causes.

Boldyreva et al. [51, 52] study what happens to the security definitions when decryption oracles can output multiple failure events. As in the blockwise adaptive setting, combining integrity and CPA security does not give CCA security. Instead, resistance against *ciphertext validity attacks* (CVA), where multiple error symbols are taken into account, is required. Then, to re-establish CCA security, CVA security and integrity under multiple error messages are needed.

Boldyreva et al. conclude that designers ideally should *"consider the possibility that their schemes might leak more than simple decryption failures."* In other words, allowing multiple decryption failures also jeopardizes the requirement that plaintext only be output on successful verification. Aside from unintentionally being leaked via error symbols, there are settings where releasing plaintext before verification is desirable. For example, it is necessary if there is not enough memory to store the entire plaintext [78] or because real-time requirements would otherwise not be met [49, 169]. Even beyond these settings, using dedicated schemes secure against the release of unverified plaintext can increase efficiency. For instance, to avoid releasing unverified plaintext into a device with insecure memory [168], the two-pass Encrypt-then-MAC composition can be used: a first pass to verify the MAC, and a second to decrypt the ciphertext. However, a single pass AE scheme suffices if it is secure against the release of unverified plaintext.

In this chapter we explore definitions for AE security when releasing unverified plaintext (RUP) is inevitable. We present the results from our paper at Asiacrypt 2014 [12] within the *subtle AE* framework of Barwell et al. [21] from IMACC 2015, where any type of leakage from the decryption oracle is modelled. Relative to Barwell et al. [21] and Boldyreva et al. [51, 52], RUP sacrifices some generality to be able to focus on what happens to the constructions presented in Chapter 6, although the definitions in this chapter are presented in full generality.

7.1 Subtle Security Definitions

As is the case in the conventional setting, AE schemes should ideally provide both confidentiality and integrity when the decryption oracle leaks. Security when the decryption oracle leaks information can be naturally defined by giving adversaries access to a leakage function

$$\Lambda : \mathsf{K} \times \mathsf{IV} \times \mathsf{C} \to \{\top\} \cup \mathsf{L}\,, \tag{7.1}$$

where L and $\{\top\}$ are disjoint, and Λ is fixed to be deterministic and stateless.

We distinguish the conventional settings from the so-called *subtle* setting with the postfix "Λ". The subtle security definitions are identical to the conventional security definitions, except the adversaries are given access to Λ. We give Λ-CCA as an example.

Definition 7.1.1 (Subtle CCA Confidentiality)**.** Let $\mathsf{P} = \mathsf{X}^*$ and let $\$: \mathsf{P} \to \mathsf{P}$ be a tweakable length-preserving URB with tweak space IV. Then the Λ-CCA-advantage of an adversary $\mathbf{A}$ against AE scheme $(\mathsf{Aenc}, \mathsf{Adec})$ is given by

$$\Lambda\text{-}\mathsf{CCA}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K^{(\cdot)}, \mathsf{Adec}_K, \Lambda_K \,;\, \mathsf{Aenc}_K^{(\cdot)} \circ \$^{(\cdot)}, \mathsf{Adec}_K, \Lambda_K)\,, \tag{7.2}$$

where $K \xleftarrow{\$} \mathsf{K}$, and $\mathbf{A}$ may not use the output of an $\mathcal{O}_1$ query as the input to an $\mathcal{O}_2$ query.

We skip the formality of writing down explicitly what happens in each IV setting, which can be done analogously to the conventional setting, as in Chapter 4.

All of the subtle definitions imply their conventional counterparts. Depending upon Λ, the reverse implications might not be true: if Λ leaks nothing, then the conventional definitions coincide with the subtle definitions, but if Λ leaks, for example, the key, then there is a clear separation between the two.

As explained by Barwell et al. [21], combining Λ-CPA confidentiality and Λ-Int integrity achieves Λ-CCA confidentiality.

Theorem 12. *Let* $(\mathsf{Aenc}, \mathsf{Adec})$ *be an AE scheme with leakage function* Λ*. Then for any* Λ-CCA*-adversary* $\mathbf{A}$

$$\Lambda\text{-}\mathsf{CCA}(\mathbf{A}) \le \Lambda\text{-}\mathsf{Int}\Big(\mathbf{A}\Big) + \Lambda\text{-}\mathsf{Int}\Big(\mathbf{A}(\circ\$, \cdot, \cdot)\Big) + \Lambda\text{-}\mathsf{CPA}\Big(\mathbf{A}(\cdot, \perp, \cdot)\Big)\,, \tag{7.3}$$

where $\$$ *is the URB from the* $(\mathsf{Aenc}, \mathsf{Adec})$ Λ-CPA*-definition.*

The proof is identical to the proof of Theorem 1 with Λ_K added to all the distinguishing bounds, and holds in all IV settings.

The definitions presented here differ in some ways from those of Barwell et al. [21]. We do not assume that the schemes are *tidy*, meaning that encryption and decryption are inverses of each other. Furthermore, our ideal oracles follow the real-or-random style, rather than the random bits style.

7.2 Is It Safe to Use Subtly Secure Schemes?

It is clear that if a subtly secure scheme is used in the conventional setting, without leakage, then security is maintained. Furthermore, from an abstract point of view the subtle security definitions provided in Section 7.1 seem natural, since they are just the conventional security definitions with the addition of Λ. Yet it remains difficult to judge whether the subtle security definitions correspond to what one would consider security when the decryption oracle leaks, since there is little connection with intuition. Extending Goldwasser and Micali's confidentiality intuition to the subtle scenario, what one would like to have is the following:

> whatever is efficiently computable about the plaintext given the ciphertext *and leakage function*, is also efficiently computable without the ciphertext and leakage function.

In other words, the leakage function should not contribute to the adversary's advantage, which is not immediately clear from the Λ-CCA and Λ-CPA definitions.

One way of formalizing this intuition is to have adversaries attempt to distinguish Aenc and Λ from Aenc and a dummy algorithm, Sim. The task of Sim is to mimic the behavior of Λ, without access to the key nor Aenc. If there exists such a Sim, then whatever advantage the adversary gets by interacting with Aenc and Λ, it could get by interacting with Aenc and Sim. Since Sim is as useless as an adversary without the key, Λ is useless as well.

This definition can be formalized via what we call *leakage simulatability*[1], capturing the idea that it is possible to simulate the leakage function Λ without access to the key.

Definition 7.2.1 (Leakage Simulatability)**.** Let Sim be an algorithm, called a Λ-simulator, which is allowed to maintain state across invocations. The LS-advantage of adversary $\mathbf{A}$ relative to Sim and (Aenc, Λ) is

$$\mathsf{LS}^{\mathsf{Sim}}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_K \,;\, \mathsf{Aenc}_K, \mathsf{Adec}_K, \mathsf{Sim}) \ , \tag{7.4}$$

[1] Note that this definition is not related to the study of leakage resilience [74].

where $K \xleftarrow{\$} \mathsf{K}$.

As Barwell et al. [21] observe, if Λ is simulatable, then the Λ-simulator does not have to be anything special: it can be implemented via Λ using an independent key. Concretely, LS is equivalent to *leakage independence*, meaning encryption and leakage under the same key are only related to each other as much as encryption and leakage under different keys. The corresponding definition by Barwell et al. [21] is called error simulatability.

Definition 7.2.2 (Leakage Independence)**.** Let $\mathbf{A}$ be a distinguisher accepting two oracles, then the LI advantage of $\mathbf{A}$ relative to (Aenc, Λ) is

$$\mathsf{LI}(\mathbf{A}) \stackrel{\text{def}}{=} \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_K \,;\, \mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_L) \ , \tag{7.5}$$

where $K, L \xleftarrow{\$} \mathsf{K}$ are independent.

The following two theorems establish equivalence of leakage simulatability and leakage independence.

Theorem 13 (Leakage Simulatability Implies Independence)**.** *Let* $(\mathsf{Aenc}, \mathsf{Adec})$ *be an AE scheme with leakage function* Λ *and* Λ*-simulator* Sim*. Let* $\mathbf{A}$ *be an* LI*-adversary, then*

$$\mathsf{LI}(\mathbf{A}) \le \mathsf{LS}^{\mathsf{Sim}}(\mathbf{A}) + \mathsf{LS}^{\mathsf{Sim}}(\mathbf{A}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \cdot)) \,. \tag{7.6}$$

Proof. By the triangle inequality,

$$\mathsf{LI}(\mathbf{A}) = \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_K \,;\, \mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_L) \tag{7.7}$$

$$\le \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_K \,;\, \mathsf{Aenc}_K, \mathsf{Adec}_K, \mathsf{Sim}) \tag{7.8}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \mathsf{Sim} \,;\, \mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_L) \,. \tag{7.9}$$

The first term is $\mathsf{LS}^{\mathsf{Sim}}(\mathbf{A})$. Furthermore, note that extractor Sim and Λ_L are independent of $(\mathsf{Aenc}_K, \mathsf{Adec}_K)$, hence applying Proposition 2.6.3

$$\underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \mathsf{Sim} \,;\, \mathsf{Aenc}_K, \mathsf{Adec}_K, \mathsf{Adec}_L) \le \underset{\mathbf{A}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \cdot)}{\Delta}(\mathsf{Sim} \,;\, \mathsf{Adec}_L) \,. \tag{7.10}$$

Since $\mathbf{A}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \cdot)$ can be viewed as an LS-adversary,

$$\underset{\mathbf{A}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \cdot)}{\Delta}(\mathsf{Sim} \,;\, \mathsf{Dec}_L) \le \mathsf{LS}^{\mathsf{Sim}}(\mathbf{A}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \cdot)) \,, \tag{7.11}$$

therefore

$$\mathsf{LI}(\mathbf{A}) \leq \mathsf{LS}^{\mathsf{Sim}}(\mathbf{A}) + \mathsf{LS}^{\mathsf{Sim}}(\mathbf{A}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \cdot)) \ . \tag{7.12}$$

□

Theorem 14 (Leakage Independence Implies Simulatability)**.** *Let* $(\mathsf{Aenc}, \mathsf{Adec})$ *be an AE scheme with leakage function* Λ*. Then for the* Λ*-simulator* $\mathsf{Sim} \stackrel{\text{def}}{=} \Lambda_L$ *with* $L \xleftarrow{\$} \mathsf{K}$ *it is the case that for any* LS*-adversary* $\mathbf{A}$*,*

$$\mathsf{LS}^{\mathsf{Sim}}(\mathbf{A}) = \mathsf{LI}(\mathbf{A}) \, . \tag{7.13}$$

Proof. The equality follows by definition:

$$\mathsf{LS}^{\mathsf{Sim}}(\mathbf{A}) = \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_K \, ; \, \mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_L) = \mathsf{LI}(\mathbf{A}) \, . \tag{7.14}$$

□

If a scheme is leakage independent and CCA-secure, then it is Λ-CCA-secure, as shown in the following theorem. The reason this is true is that a Λ-CCA-adversary $\mathbf{A}$ against a leakage independent scheme could be viewed as a CCA-adversary $\mathbf{A}(\cdot, \cdot, \Lambda_L)$ which simply simulates the leakage independently and runs the Λ-CCA adversary.

Theorem 15. *Let* $(\mathsf{Aenc}, \mathsf{Adec})$ *be an AE scheme with leakage function* Λ*, then for any* Λ-CCA*-adversary* $\mathbf{A}$

$$\Lambda\text{-}\mathsf{CCA}(\mathbf{A}) \leq \mathsf{CCA}(\mathbf{A}(\cdot, \cdot, \Lambda_L)) + \mathsf{LI}(\mathbf{A}) + \mathsf{LI}(\mathbf{A}(\circ\$, \cdot, \cdot)) \, . \tag{7.15}$$

Proof. Using the triangle inequality we get

$$\Lambda\text{-}\mathsf{CCA}(\mathbf{A}) = \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_K \, ; \, \mathsf{Aenc}_K \circ \$, \mathsf{Adec}_K, \Lambda_K) \tag{7.16}$$

$$\leq \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_K \, ; \, \mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_L) \tag{7.17}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K, \mathsf{Adec}_K, \Lambda_L \, ; \, \mathsf{Aenc}_K \circ \$, \mathsf{Adec}_K, \Lambda_L) \tag{7.18}$$

$$+ \underset{\mathbf{A}}{\Delta}(\mathsf{Aenc}_K \circ \$, \mathsf{Adec}_K, \Lambda_L \, ; \, \mathsf{Aenc}_K \circ \$, \mathsf{Adec}_K, \Lambda_K) \, . \tag{7.19}$$

The first term is exactly $\mathsf{LI}(\mathbf{A})$, the second term is $\mathsf{CCA}(\mathbf{A}(\cdot, \cdot, \Lambda_L))$, and the third term is $\mathsf{LI}(\mathbf{A}(\circ\$, \cdot, \cdot))$. □

The converse is not true: if the AE scheme and leakage function both leak one bit of a large key, then they will most likely maintain confidentiality, whereas it

will be easy to determine if the AE scheme and leakage are independent or not. This means that even if a scheme is Λ-CCA-secure, the leakage function could actually help the adversary. In order for a scheme to achieve confidentiality as described by the intuition provided in the beginning of the section, it should satisfy leakage independence on its own.

7.3 Releasing Unverified Plaintext

The leakage function Λ in the subtle AE framework models is left unspecified, allowing one to model a wide range of scenarios. We focus on the case where Λ releases unverified plaintext when Adec returns ⊥. In other words, we assume there exists an algorithm $\Lambda : \mathsf{K} \times \mathsf{IV} \times \mathsf{C} \to \{\top\} \cup \mathsf{P}$ with $\top \notin \mathsf{P}$, where if $\mathsf{Adec}_K(C) = P \in \mathsf{P}$ then $\Lambda_K(C) = \top$, and if $\mathsf{Adec}_K(C) = \bot$, then $\Lambda_K(C) = P$ for some $P \in \mathsf{P}$ which would have been output if $\mathsf{Adec}_K(C)$ did not output its error symbol. Depending upon the scheme, such a Λ might not make sense, although many practical AE schemes can be viewed as having separate encryption and authentication processes, allowing one to extract such a Λ.

Example 7.3.1 (GCM in RUP Setting)**.** GCM is an encrypt-then-MAC style AE scheme, which means it looks like

$$\mathsf{Aenc}_K^N(P) = \mathsf{Tag}_K^N(\mathsf{Enc}_K^N(P)) \tag{7.20}$$

$$\mathsf{Adec}_K(N, C) = \begin{cases} \mathsf{Dec}_K(N, C) & \text{if } \mathsf{Ver}_K(N, C) = 1 \\ \bot & \text{otherwise}\,. \end{cases} \tag{7.21}$$

The encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is CTR mode, and the authenticator $(\mathsf{Tag}, \mathsf{Ver})$ is a polynomial-based MAC. In the conventional setting GCM outputs only ⊥ if verification is faulty. In the RUP setting adversaries are also given access to Λ, which for the case of GCM is defined as

$$\Lambda_K(N, A, C) = \begin{cases} \top & \text{if } \mathsf{Ver}_K(N, A, C) = 1 \\ \mathsf{Dec}_K(N, C) & \text{otherwise}\,, \end{cases} \tag{7.22}$$

meaning decryption occurs anyway if verification fails. ◀

Example 7.3.2 (Encode-then-Encipher)**.** The encode-then-encipher construction from Section 6.4.3, which uses a tweakable cipher to achieve AE, checks integrity to see if decryption results in a plaintext with a particular constant appended. Its leakage function releases the decrypted plaintext regardless of whether the decoding succeeded or not:

$$\Lambda_K(N, C) = \begin{cases} \top & \text{if } \mathsf{D}_K^N(C) = (P, P_0) \\ \mathsf{D}_K^N(C) & \text{otherwise}\ . \end{cases} \tag{7.23}$$

◀

7.3.1 RUP Insecurity

Since Λ outputs the decryption of the ciphertext regardless of verification, by giving adversaries access to Adec_K and Λ_K, they effectively have access to the decryption part of the underlying encryption scheme. Most AE schemes are designed to only satisfy CPA as an encryption scheme, since combining CPA with Int allows one to achieve CCA-security, and in the interest of efficiency, many AE schemes *only* satisfy CPA-security making them immediately vulnerable in the Λ-CCA setting. For example, GCM in the RUP setting effectively turns into CTR mode, allowing one to mount the CCA-attack described in Chapter 5.

Even if the underlying encryption scheme is CCA-secure, one might not achieve RUP security if authentication is done separately from decryption. This is, for example, the case in AE schemes where the ciphertext is computed using some length-preserving bijective function, and then a "tag" is appended to the ciphertext. Such schemes achieve AE security because the tag prevents all ciphertexts from being valid, but if the tag is no longer checked, then RUP confidentiality cannot be achieved. Concretely, if $(\mathsf{Aenc}, \mathsf{Adec})$ is an AE scheme such that

$$\mathsf{Aenc}_K^N(A, P) = \mathsf{E}_K^N(A, P) \parallel \mathsf{F}_K^N(A, P) \ , \tag{7.24}$$

where E_K is length-preserving, i.e. $\left|\mathsf{E}_K^N(A, P)\right| = |P|$. Then one can always encrypt arbitrary (A, P) receiving $C_1 \| C_2$ with $|C_1| = |P|$, modify C_2, which is the part corresponding to $\mathsf{F}_K^N(A, P)$, thereby creating a ciphertext $C_1 \| C_2'$, and ask for $\Lambda_K(N, C_1 \| C_2')$. When interacting with Aenc_K the output of $\Lambda_K(N, C_1 \| C_2')$ will have P as a prefix, and when interacting with $\mathsf{Aenc}_K \circ \$$, the output of Λ_K will be independent of P, resulting in a distinguishing attack.

For integrity, there are no obvious ways that constructions could fail to provide security in the RUP setting. Nevertheless, several AE schemes become insecure if unverified plaintext is released. In Proposition 7.3.1, we demonstrate an attack against OCB [155].

The strategy of the attack is similar to that of Bellare and Micciancio on the XHASH hash function [31]. The attack works by first querying the encryption oracle under nonce N to get a valid ciphertext and tag pair. Then, two decryption queries are made under the same nonce N. Using the resulting plaintexts a system of linear equations is set up, which when solved will give a forgery with high probability.

Proposition 7.3.1. *For OCB, for all $\ell \geq n$ there exists an adversary $\mathbf{A}$ such that*

$$\Lambda\text{-}\mathsf{Int}(\mathbf{A}) \geq 1 - 2^{n-\ell} \ , \tag{7.25}$$

where $\mathbf{A}$ makes one encryption query and two decryption queries, each consisting of ℓ blocks of n bits. Then, the adversary solves a system of linear equations in $GF(2)$ with n equations and ℓ unknowns.

Proof. We start by describing OCB for messages which have a length which is a multiple of the block size. For our purposes it suffices to describe OCB in terms of a tweakable URP, since the attack is independent of the underlying block cipher.

Let $\Pi = (\mathsf{Aenc}, \mathsf{Adec})$ denote OCB operating only on full message blocks. Let $\{\alpha_i^N, \beta_i^N, \gamma_i^N\}$ be a family of URPs over $\{0,1\}^n$ with tweaks given by the subscript i and superscript N, then OCB is defined as

$$\mathsf{Aenc}_K(N, M_1 M_2 \cdots M_\ell) = (N, C_1 C_2 \cdots C_\ell, T) \ , \tag{7.26}$$

where

$$C_i = \alpha_i^N(M_i) \qquad \text{for } 1 \leq i < \ell \ , \tag{7.27}$$

$$C_\ell = \beta_\ell^N(\mathrm{len}(n)) \oplus M_\ell \ , \tag{7.28}$$

$$T = \gamma_\ell^N\Big(M_1 \oplus \cdots \oplus M_\ell\Big) \ , \tag{7.29}$$

and $\mathrm{len}(n)$ is the number n represented as an n-bit string.

Given a valid plaintext-ciphertext pair, $\mathbf{A}$ makes two queries to the decryption oracle, and then solves a system of linear equations in $GF(2)$ in order to obtain a forgery.

Let $\ell \geq n$. First, $\mathbf{A}$ queries $\mathsf{Aenc}_K(N, M) = (N, C, T)$ where $M = M_1 M_2 \cdots M_\ell$ consists of ℓ blocks of n bits, and N is some fixed value. Let $C = C_1 C_2 \cdots C_\ell$ and let $Z = M_1 \oplus \cdots \oplus M_\ell$.

If $\mathbf{A}$ can create another plaintext M' with the same checksum Z by changing the message blocks $M_1, M_2, ..., M_\ell$, it has constructed a forgery because the checksum Z and therefore the tag T will be the same. The adversary is not allowed to query two encryptions under the same nonce N. However, we now show that it is possible to construct a forgery by querying the decryption oracle twice with the same nonce N and observing the unverified plaintext.

The adversary chooses $C^0 = C_1^0 C_2^0 \cdots C_\ell^0 T^0$ and $C^1 = C_1^1 C_2^1 \cdots C_\ell^1 T^1$ uniformly at random such that for each i, C_i^0, C_i^1, C_i are all distinct. The corresponding

unverified plaintexts are $\Lambda_K(N, C^0, T^0) = M_1^0 M_2^0 \cdots M_\ell^0$ and $\Lambda_K(N, C^1, T^1) = M_1^1 M_2^1 \cdots M_\ell^1$. To construct a plaintext $M' = M_1^{x_1} M_2^{x_2} \cdots M_\ell^{x_\ell}$ with the same checksum as M, the adversary has to find $x_1, x_2, \cdots, x_\ell \in \mathrm{GF}(2)$ such that

$$Z = \oplus_{i=1}^{\ell} \big(M_i^0 x_i \oplus M_i^1 (x_i \oplus 1) \big) \ , \tag{7.30}$$

where $x_i = 1$ corresponds to selecting M_i^0, and $x_i = 0$ to selecting M_i^1 as the ith message block of M'. This expression can be converted into n equations, one for every bit j:

$$Z[j] = \oplus_{i=1}^{\ell} \big(M_i^0[j] x_i \oplus M_i^1[j](x_i \oplus 1) \big) \qquad \text{for } j = 0, 1, \ldots, n-1 \ , \tag{7.31}$$

where $X[j]$ selects jth bit of X, with $j = 0$ corresponding to the least significant bit.

This is a system of linear equations in $GF(2)$ with n equations and ℓ unknowns, for which a solution can be found using Gaussian elimination. The probability that this system of equations has a solution, is at least $1 - 2^{n-\ell}$ [31, App. A]. Because $\mathsf{Aenc}_K(N, M') = (N, C', T)$ with $C' = C_1^{x_1} C_2^{x_2} \cdots C_\ell^{x_\ell}$ and $C' \neq C$, the adversary can output (N, C', T) as a forgery. □

7.3.2 RUP-Secure Constructions

Currently, the only known method of achieving a RUP-secure scheme is to use the encode-then-encipher approach with a tweakable cipher, such as the solutions presented by Bellare and Rogaway [34], Desai [69], and Shrimpton and Terashima [165]. These constructions are already CCA-secure without an integrity check, meaning even if Λ_K outputs plaintext, adversaries will not gain any useful information to perform a confidentiality attack. Furthermore, tweakable ciphers have strong decryption algorithms, which means that decrypting an arbitrary ciphertext will result in plaintext that is computationally indistinguishable from random. In particular, it is very unlikely that the decryption of an arbitrary ciphertext will result in a plaintext which conforms to the proper encoding, meaning integrity will be preserved as well. These arguments are formalized by Shrimpton and Terashima [165] and Hoang, Krovetz, and Rogaway [92].

Achieving just integrity in the RUP setting is possible without resorting to tweakable ciphers. Starting with any IV-based encryption scheme one can add a VIL-PRF to construct a scheme which is Λ-Int-secure, using a technique similar to MAC-then-Encrypt [32]. The idea behind the *PRF-to-IV* method is to evaluate a VIL-PRF over the input to the scheme and then to use the resulting output as an IV for the IV-based encryption scheme. Let Π =

$(\mathsf{Enc}, \mathsf{Dec})$ be an IV-based encryption scheme taking IVs from $\{0,1\}^n$, and let $\mathsf{F} : \mathsf{K} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be a VIL-PRF, then define $(\mathsf{Aenc}, \mathsf{Adec})$ as follows:

$$\mathsf{Aenc}^N_{K_1,K_2}(P) \stackrel{\text{def}}{=} (N, N', \mathsf{Enc}^{N'}_{K_2}(P)) \tag{7.32}$$

$$\text{where } N' = \mathsf{F}_{K_1}(N, P) \tag{7.33}$$

$$\mathsf{Adec}_{K_1,K_2}(N, N', C) \stackrel{\text{def}}{=} \begin{cases} P & \text{if } \mathsf{F}_{K_1}(N, P) = N' \\ \perp & \text{otherwise} \ , \end{cases} \tag{7.34}$$

$$\text{where } P = \mathsf{Dec}^{N'}_{K_2}(C)\,. \tag{7.35}$$

In this case Λ simply outputs $\mathsf{Dec}^{N'}_{K_2}(C)$ in the "otherwise" case of Adec.

Proposition 7.3.2. *Let* $(\mathsf{Aenc}, \mathsf{Adec})$ *be the PRF-to-IV method described above with corresponding leakage function* Λ*. Let* $\mathbf{A}$ *be an INT-RUP adversary for* $(\mathsf{Aenc}, \mathsf{Adec})$ *making at most* v *forgery attempts, and let* $\mathbf{B}$ *be a* VIL-PRF *adversary against* F *which runs* $\mathbf{A}$*, generates a random key* K_2*, and simulates* $(\mathsf{Enc}_K, \mathsf{Dec}_K)$ *using* K_2 *and its own oracle, and outputs* 1 *if* $\mathbf{A}$ *succeeds in constructing a forgery, and* 0 *otherwise. Then*

$$\Lambda\text{-}\mathsf{Int}(\mathbf{A}) \leq \mathsf{VIL\text{-}PRF}_{\mathsf{F}}(\mathbf{B}) + \frac{v}{2^n} \ . \tag{7.36}$$

Proof. $\mathbf{A}$ only succeeds in constructing a forgery if it is able to predict the output of F, which it can only do with probability at most $v/2^n$, assuming F approximates a VIL-URF well. □

Chapter 8

Bound Tightness

Aside from the introduction, the contents of this chapter are from our publications on LightMAC [122] and the analysis of PMAC [119]. The author of this thesis is also the main author of the two publications and, except for Appendix C, all text included here was written by the author of this thesis.

8.1 Introduction

When searching for optimal cryptographic schemes, security bounds provide an important tool for selecting the right parameters, like the key size, tag size, or block size. Security bounds capture the concept of explicitly measuring the effect of an adversary's resources on its success probability in breaking the scheme, relative to the chosen parameters. They enable one to determine how intensively a scheme can be used in a session. Therefore, reducing the impact of an adversary's resources from, say, a quadratic to a linear term, can mean an order of magnitude increase in a scheme's lifetime. Conversely, finding attacks which confirm an adversary's success rate, relative to its allotted resources, prove claims of security bound optimality.

As discussed in Chapter 5, Section 5.1, the security bound for a mode of operation using a primitive can be split into two components: the primitive's quality, and the mode's insecurity when used with an ideal primitive. Taking the CTR mode example, Theorem 5 establishes that the n-CPA-advantage of any adversary $\mathbf{A}$ against CTR mode is bounded above by

$$\mathsf{PRP}(\mathbf{B}\langle\mathbf{A}\rangle) + \mathsf{PRP}(\mathbf{B}\langle\mathbf{A}\rangle(\circ\$)) + \mathsf{n}\text{-}\mathsf{CPA}_{(\mathsf{Enc}[\pi],\mathsf{Dec}[\pi])}(\mathbf{A})\,, \tag{8.1}$$

where $\mathbf{B}\langle\cdot\rangle$ is the CTR mode reduction. This means that there are only two ways of attacking CTR mode with a block cipher: either attack the block cipher, or attack $(\mathsf{Enc}[\pi], \mathsf{Dec}[\pi])$, which is CTR mode with the ideal primitive π. To be able to make concrete guarantees on how extensively CTR mode can be used, estimates need to be given on both the PRP quality of the underlying block cipher, and the maximum n-CPA advantage possible against $(\mathsf{Enc}[\pi], \mathsf{Dec}[\pi])$.

Estimating the PRP quality of block ciphers, and the quality of any primitive in general, is a non-trivial problem. With any new primitive design the initial hypothesis is that no attack is significantly better than "brute force", where every possible key in K is tested against a known input-output pair. The hypothesis can only be tested through years of research, thereby adding evidence to its veracity, or possibly weakening the hypothesis. The duration for which a primitive can be used under a single key is determined via the most up-to-date hypothesis. For example, for the Advanced Encryption Standard block cipher using 128 bit keys, it is generally accepted that adversaries will have to take roughly 2^{127} time on average to break its PRP quality.

In contrast, estimating mode insecurity with an ideal primitive can be done more precisely. For example, Theorem 6 establishes that for any n-CPA-adversary $\mathbf{A}$ querying at most σ plaintext blocks,

$$\mathsf{n\text{-}CPA}_{(\mathsf{Enc}[\pi],\mathsf{Dec}[\pi])}(\mathbf{A}) \leq \frac{\sigma^2}{2^n}\,. \tag{8.2}$$

The theorem describes $\mathbf{A}$'s advantage purely in terms of the amount of data it sees, and ignores running time. This contrasts with finding attacks against well-designed primitives, where the best known attacks barely improve as a function of data, and running time is the dominant factor.

Using a combination of the primitive hypothesis and the mode security bound, one can estimate the maximum length of time and amount of data for which one can use a scheme until it becomes vulnerable to attacks. Designing primitives is out of the scope of this thesis, therefore henceforth we will assume that there exist well-designed primitives which can be used in modes of operation, and we do not take into account attacks against the primitive. Instead, we will look at the impact of security bounds of modes using ideal primitives.

8.2 MAC Bounds

MAC algorithms provide a good example of schemes which have been studied extensively to determine optimal bounds. A MAC's security bound is measured as a function of the number of tagging queries, q, and the largest message

length, ℓ, used before a first forgery attempt is successful. The impact of an adversary's resources, q and ℓ, on its success probability in breaking a MAC is then described via an upper bound of the form $f(q, \ell) \cdot \epsilon$, where f is a function, often a polynomial, and ϵ is a quantity dependent on the MAC's parameters. The maximum number of queries $q_{\max}$ with length $\ell_{\max}$ one can make under a key is computed by determining when $f(q_{\max}, \ell_{\max}) \cdot \epsilon$ is less than some threshold success probability. For example, if one is comfortable with adversaries which have a one in a million chance of breaking the scheme, but no more, then one would determine $q_{\max}$ and $\ell_{\max}$ via

$$f(q_{\max}, \ell_{\max}) \cdot \epsilon \leq 10^{-6} \,. \tag{8.3}$$

Given that $q_{\max}$ and $\ell_{\max}$ depend only on f, it becomes important to find the f which establishes the tightest upper bound on the success probability.

The optimality of f depends on the environment in which the MAC operates, or in other words, the assumptions made on the MAC. For instance, nonce-based MACs, such as the Wegman-Carter construction [174], can achieve bounds independent of q and ℓ. In this case, an adversary's success remains negligible regardless of q and ℓ, as long as the construction receives nonces. Therefore, determining $q_{\max}$ and $\ell_{\max}$ for Wegman-Carter MACs amounts to solving $\epsilon \ll 1$, which is true under the assumption that IVs are unique. Similarly, XOR MAC [26] with nonces achieves a security upper bound of $\epsilon = 1/2^{\tau}$, with τ the tag length in bits, which is the optimal bound for any MAC. Randomized, but stateless MACs can achieve bounds similar to stateful MACs, as shown by Minematsu [126].

In contrast, deterministic and stateless MACs necessarily have a lower bound of $q^2/2^n$, where n is the inner state size, due to a generic attack by Preneel and van Oorschot [146]. This means that for any f,

$$f(q, \ell) \cdot \epsilon \geq \frac{q^2}{2^n} \,, \tag{8.4}$$

hence any deterministic, stateless MAC must use fewer than $2^{n/2}$ tagging queries per key.

Given this lower limit on f, one would perhaps expect to find schemes for which the proven upper bound is $q^2/2^n$. Yet many deterministic, stateless MACs have upper bounds including an ℓ-factor. Block cipher based MACs, such as CBC-MAC [27], OMAC [98], and PMAC [47], were originally proven with an upper bound on the order of $q^2\ell^2/2^n$, growing quadratically as a function of ℓ relative to a fixed block size n. Much effort went to improving the bounds to a linear dependence on ℓ, resulting in bounds of the form $q^2\ell/2^n$; see Table 8.1 for a list of modes with their dependence on ℓ.

Table 8.1: The table below contains the coefficients of the powers of ℓ contained in the security bounds for adversaries making q queries of length ℓ, with block size n bits. References are to papers proving the bounds. In the bound for EMAC, the function $d'(\ell)$ has been replaced by ℓ.

Mode	1	ℓ	ℓ^2	ℓ^3	ℓ^4
3kf9 [183]	$\frac{4q}{2^n} + \frac{4q^3}{2^{2n}}$	$\frac{4q}{2^n} + \frac{4q^3}{2^{2n}}$	$\frac{2q^3}{2^{2n}}$	$\frac{4q^3}{2^{2n}}$	
CBC-MAC [33]		$\frac{12q^2}{2^n}$			$\frac{64q^2}{2^{2n}}$
EMAC [33]		$\frac{q^2}{2^n}$			$\frac{32q^2}{2^{2n}}$
OMAC [131]		$\frac{5q^2}{2^n}$			$\frac{8q^2}{2^{2n}}$
PMAC [138]	$\frac{-3.5q^2}{2^n}$	$\frac{5q^2}{2^n}$			
PMAC_Plus [179]		$\frac{3q}{2^n}$		$\frac{27q^3}{2^{2n}}$	
PMACX [185] (m=14,l=12)	$\frac{72+1.5q^2}{2^n} + \frac{576q^2}{2^{2n}}$	$\frac{576q^2}{2^{2n}}$	$\frac{144q^2}{2^{2n}}$		
PMAC with Parity [180]	$\frac{q^2}{2^n}$		$\frac{q^2}{2^{2n}}$		
Sum of CBCs [178]					$\frac{12q^3}{2^{2n}}$

The dependence on ℓ and the block size n can create issues when n is small. As shown in Table 8.2, block sizes range from 128 down to 32 bits. With a 32 bit block size and a guarantee that adversaries do not forge with probability more than one in a million, one gets a restriction of the form

$$\frac{q^2\ell}{2^{32}} \leq \frac{1}{2^{20}} \quad \text{or} \quad q^2\ell \leq 2^{12}\,, \tag{8.5}$$

meaning 64 one-block messages can be tagged under the same key. But what if the messages are longer than one block? With conventional MACs only 32 four-block messages can be tagged, corresponding to $32 \cdot 2^2 \cdot 32 = 2^{12}$ bits, or 512 Bytes of data per key. If the messages are sixteen blocks long, only 16 messages can be tagged, which is $16 \cdot 2^4 \cdot 32 = 2^{13}$ bits, or 1 KiB of data per key. Figure 8.1 displays how much data the various modes from Table 8.1 can process per key, when the threshold success probability is set to $1/2^{20}$.

Table 8.2: Supported block sizes are often small, and can be as low as 32 bits.

Block size (bits)	32	48	64	80	96	128	256
3DES [20]			×				
AES [66]						×	
CLEFIA [163]						×	
DESLX [113]			×				
Fantomas [86]						×	
HIGHT [95]			×				
ITUbee [106]				×			
KLEIN [84]			×				
KATAN [56]	×	×	×				
LBlock [176]			×				
LED [88]			×				
LEA [94]						×	
mCrypton [115]			×				
Mysterion [102]						×	×
Noekeon [65]						×	
Piccolo [162]			×				
PRESENT [48]			×				
PRIDE [6]			×				
PRINCE [54]			×				
RC5 [151]	×		×			×	
Rectangle [184]			×				
Rijndael [66]						×	×
RoadRunneR [22]			×				
Robin [86]						×	
SEA [166]					×		
SIMECK [177]	×	×	×				
Simon [23]	×	×	×		×	×	
Speck [23]	×	×	×		×	×	
TWINE [167]			×				
XTEA [140]			×				
Zorro [82]						×	

For certain deterministic, stateless schemes the dependence on ℓ has been proven to be necessary. Dodis and Pietrzak [70] point out that this is the case for polynomial based MACs, and try to avoid the dependence by introducing randomness. Pietrzak [144] notes that the EMAC bound must depend on ℓ. Gazi, Pietrzak, and Rybár [80] give an attack on NMAC showing its dependence on ℓ. Nevertheless, there are no known generic attacks establishing a lower

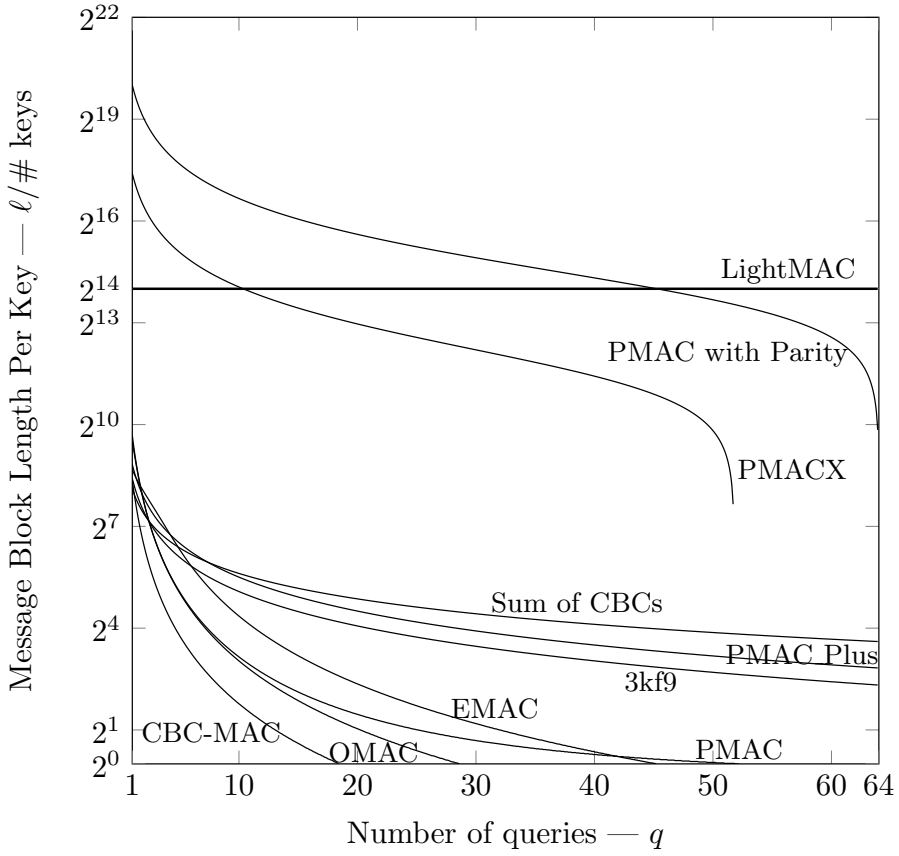


Figure 8.1: A plot of message block lengths per key versus the number of queries that can be made in order to achieve the threshold success probability of 2^{-20}. In other words, if (x, y) is a point on the graph, then $x \cdot y$ represents the number of blocks that can be processed per key. The block size is set to 32 bits.

bound of the form $\ell^\epsilon/2^n$ for any $\epsilon > 0$.

In certain cases the bounds in Table 8.1 can be improved. For example, for EMAC, Pietrzak [144] proved that if $\ell \leq 2^{n/8}$ and $q \geq \ell^2$, then the bound's order of growth is independent of ℓ. The proven bound is

$$128 \cdot \frac{q^2\ell^8}{2^{2n}} + 16 \cdot \frac{q^2}{2^n} + \frac{q(q-1)}{2^{n+1}}\,. \tag{8.6}$$

Note that the condition on ℓ means that EMAC's bound is not truly independent of ℓ. For the sum of CBCs, Yasuda [178] also showed that if $\ell \leq 2^{2n/5}$, the advantage becomes $\frac{40\ell^3q^3}{2^{2n}}$. Rogaway [153] has shown that the dependence on ℓ disappears if you consider a version of PMAC with an ideal tweakable block cipher.

8.3 LightMAC

We present a MAC mode, LightMAC, which enables one to tag much longer messages than typically possible. LightMAC is depicted in Figure 8.2 and Algorithm 1.

The security upper bound for LightMAC is

$$(1+\epsilon) \cdot \frac{q^2}{2^n} \quad \text{where } \epsilon \in O\left(\frac{1}{2^{n/2}-1}\right)\,, \tag{8.7}$$

which is independent of the message length (see Section 8.3.3). In other words, with a 32 bit block size, and setting the message-length parameter s to 16, roughly 64 messages can be tagged with length up to 2^{15} blocks. Note that keys are used most efficiently when the messages are as long as possible: up to $64 \cdot 2^{15} \cdot 32 = 2^{26}$ bits, or *8 MiB* of data can be tagged per key. LightMAC uses two independent keys, but even after normalizing by the number of keys, the amount of data processed per key is still 4 MiB, a significant improvement over 1 KiB.

Figure 8.1 compares LightMAC to the other published modes from Table 8.1. The figure shows that LightMAC starts with a factor 2^4 improvement over many of the modes, which grows to roughly 2^{10} as the number of queries increases. Modes such as PMAC with Parity and PMACX were designed to handle long message lengths and offer competitive bounds, at the cost of increased design complexity. LightMAC's advantage over these modes is its simplicity and low overhead.

Like PMAC [47], LightMAC allows block cipher calls to be made in parallel, but unlike PMAC, LightMAC is based on Bernstein's *protected counter sum* [37], and hence should not suffer from patent issues (PMAC patent [152]).

A disadvantage of LightMAC is that its rate is low. In order to tag messages of length up to $2^{n/2-1}$ blocks, $n/2$ bits of the block must be sacrificed for a counter, hence two block cipher calls must be called per block of data. However, the rate can be improved: if the maximum message length that will be communicated is known to be less than $2^s(n-s)$ bits, then the rate can be set to $(n-s)/n$ blocks per block cipher call. For example, using a 32 bit block cipher, if the message lengths are less than 2^9 blocks, then the rate can be set to $2/3$ blocks per call. Therefore, unlike other modes, LightMAC can be optimized according to the application: the shorter the messages, the more efficient LightMAC is, while allowing the same number of message to be queried.

8.3.1 Design

Yasuda [180] explained the basic idea for LightMAC in his paper's introduction, which can be viewed as an adaptation of Bernstein's protected counter sum [37] using block ciphers. Recall from Example 5.6.7 that the protected counter sum maps $M_1, M_2, \ldots, M_\ell$ using a PRF $\varphi : \mathsf{K} \times \mathbf{N} \times \mathsf{X} \to \mathsf{Y}$ to

$$\varphi_K\left(0\ ,\ \bigoplus_{i=1}^{\ell} \varphi_K(i, M_i)\right)\ . \tag{8.8}$$

Due to its use of PRFs, the protected counter sum achieves a security bound which is independent of the message length, since the XOR of independent, uniformly distributed random variables is still uniformly distributed.

However, trying to use a block cipher in the protected counter sum, one runs into difficulties. If one were to use a block cipher directly as a PRF, then the security bound would incur a loss of $q^2\ell^2/2^{n+1}$ due to a necessary application of the PRP-PRF switch (Lemma 2). Alternatively one could construct a PRF from a block cipher and then use it in the protected counter sum, by, for example, truncating the block cipher output or XORing together two block cipher calls per PRF call. Yet truncating the output of a 32 bit block cipher would result in an incredibly small output, thereby increasing chances of constructing a forgery, and XORing together two block cipher calls would result in an inefficient scheme.

Instead, LightMAC uses an independent key for the last block cipher call, and we prove directly that using a block cipher results in a bound which is independent of the message length.

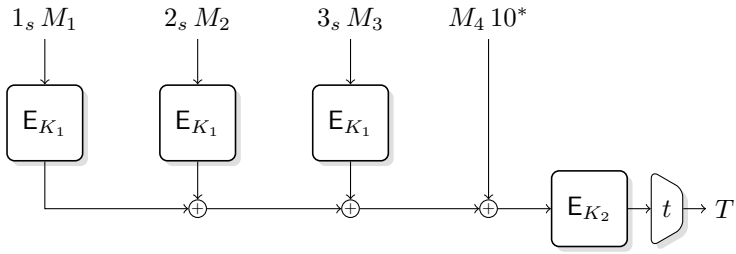


Figure 8.2: LightMAC evaluated on a message $M_1\, M_2\, M_3\, M_4 \xleftarrow{n-s} M$. The rounded squares represent block cipher calls and the trapezium is truncation to t bits.

8.3.2 Specification

Let $\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Let s and t be integers not greater than $n/2$ and n, respectively. For an integer $1 \leq i \leq 2^s$, let i_s represent some s-bit constant with the property that if $1 \leq i < j \leq 2^s$ then $i_s \neq j_s$. For example, i_s could be an s-bit representation of the integer i, or the ith s-bit Gray code. LightMAC accepts two independent and uniformly generated keys K_1 and K_2 from $\{0,1\}^k$, and a message M of length at most $2^s(n-s)$ bits. LightMAC produces an output of length t bits.

Figure 8.2 and Algorithm 1 depict how the output is produced. In Figure 8.2 and Algorithm 1, $M_1 M_2 \cdots M_\ell \xleftarrow{r} M$ represents splitting M into r-bit blocks with the length of the last block, M_ℓ, being anywhere from zero to $r-1$ bits. Also, given a block length n, concatenation of 10^* to a string means appending a one followed by the minimum number of zeros to make the total string length a multiple of n bits.

LightMAC can be used as either a VIL-PRF or a MAC. When used as a VIL-PRF, LightMAC is fully described by Algorithm 1. When used as a MAC, tags are generated using Algorithm 1, and verification of a message-tag pair (M, T) is done by comparing LightMAC (M) with T: if the two are equal, verification succeeds, otherwise not.

The parameters of LightMAC are the integers s and t, the representation of i_s, and the block cipher E, which implicitly fixes k and n. The parameters must be agreed upon before a session starts, and remain constant during.

Algorithm 1: LightMAC $_{K_1,K_2}(M)$.

Input: $K_1, K_2 \in \{0,1\}^k$, $M \in \{0,1\}^{\leq 2^s(n-s)}$
Output: $T \in \{0,1\}^t$
$V \leftarrow 0^n \in \{0,1\}^n$
$M_1 M_2 \cdots M_\ell \xleftarrow{n-s} M$
for $i = 1$ **to** $\ell - 1$ **do**
  $V \leftarrow V \oplus E_{K_1}(i_s \, M_i)$
$V \leftarrow V \oplus (M_\ell \, 10^*)$
$T \leftarrow \lfloor E_{K_2}(V) \rfloor_t$
return T

8.3.3 Security

The theorems in this section assume that E_{K_1} and E_{K_2} have been replaced by independent URPs π_1 and π_2 as discussed in Chapter 5.

LightMAC as a VIL-PRF.

Theorem 16. *Let* $\mathbf{A}$ *be a* VIL-PRF*-adversary against LightMAC making at most* q *queries of length at most* $2^s(n-s)$ *bits, then*

$$\mathsf{VIL\text{-}PRF}_{LightMAC}(\mathbf{A}) \leq \left(1 + \frac{1}{2^{n/2}-1} + \frac{1}{2(2^{n/2}-1)^2}\right) \cdot \frac{q^2}{2^n}\,, \tag{8.9}$$

where n *is the block size in bits.*

Proof. We replace π_2 with a URF ϕ using Lemma 2, at a cost of $q^2/2^{n+1}$ in advantage. The VIL-PRF we are left with is

$$\Phi(M) = \phi\left(M_\ell 10^* \oplus \bigoplus_{i=1}^{\ell-1} \pi_1(i_s M_i)\right)\,, \tag{8.10}$$

which is LightMAC instantiated with π_1 and ϕ, and

$$\mathsf{VIL\text{-}PRF}_{LightMAC}(\mathbf{A}) \leq \mathsf{VIL\text{-}PRF}_{\Phi}(\mathbf{A}) + \frac{q^2}{2^{n+1}}\,. \tag{8.11}$$

Let F denote the function contained in the call to ϕ in Equation (8.10). Then, as long as F's outputs are distinct, each input to ϕ is unique, meaning Φ will

be indistinguishable from a VIL-URF. In other words,

$$\mathsf{VIL\text{-}PRF}_{\Phi}(\mathbf{A}) \leq \sum_{i<j} \mathbf{P}\left[F(M^i) = F(M^j)\right] \leq \frac{q^2}{2} \max_{M^i \neq M^j} \mathbf{P}\left[F(M^i) = F(M^j)\right], \tag{8.12}$$

where M^i for $i = 1, \ldots, q$ are the messages queried by $\mathbf{A}$. The maximum on the right hand side is computed in Section 8.3.4, resulting in the bound

$$\mathsf{VIL\text{-}PRF}_{\Phi}(\mathbf{A}) \leq \frac{q^2}{2} \cdot \frac{1}{2^n - 2^{s+1} + 1}. \tag{8.13}$$

Therefore, using the fact that $s \leq n/2$, we have

$$\mathsf{VIL\text{-}PRF}_{LightMAC} \leq \frac{q^2}{2^{n+1}} + \frac{q^2}{2} \cdot \frac{1}{2^n - 2^{s+1} + 1} \tag{8.14}$$

$$\leq \frac{q^2}{2^n}\left(1 + \frac{1}{2^{n/2} - 1} + \frac{1}{2(2^{n/2} - 1)^2}\right), \tag{8.15}$$

giving us our desired bound. □

LightMAC as a MAC.

Theorem 17. *The* a-Int-*advantage against LightMAC of any adversary* $\mathbf{A}$ *making at most* q *tagging queries and* v *verification queries of length at most* $2^s(n - s)$ *bits, is bounded above by*

$$\left(1 + \frac{2}{2^{n/2} - 1} + \frac{1}{(2^{n/2} - 1)^2}\right) \cdot \left(\frac{q^2}{2^n} + \frac{v}{2^t}\right), \tag{8.16}$$

where n *is the block size in bits.*

Proof. As a MAC, LightMAC follows the hash-then-encrypt paradigm with the function F from Section 8.3.4 as the "hash" part, hence applying Theorem 8 we get an upper bound of

$$\left(1 + \frac{2}{2^{n/2} - 1} + \frac{1}{(2^{n/2} - 1)^2}\right) \cdot \left(\frac{q^2}{2^n} + \frac{v}{2^t}\right). \tag{8.17}$$

□

8.3.4 Collision Probability of F

Proposition 8.3.1. *Let $m = 2^s(n-s)$. Let $M_1 M_2 \cdots M_\ell \xleftarrow{n-s} M$ for $M \in \{0,1\}^{\leq m}$, and define F to be*

$$F(M) = M_\ell 10^* \oplus \bigoplus_{i=1}^{\ell-1} \pi(i_s \, M_i)\,, \tag{8.18}$$

where π is a URP over $\{0,1\}^n$, then the probability that two distinct messages $M^1, M^2 \in \{0,1\}^{\leq m}$ collide is

$$\mathbf{P}\left[F(M^1) = F(M^2)\right] \leq \frac{1}{2^n - \ell_1 - \ell_2 + 1}\,, \tag{8.19}$$

where ℓ_i is the length of M^i in $(n-s)$-bit blocks rounded up.

Proof. The equation $F(M^1) = F(M^2)$ can be rewritten as

$$\bigoplus_{i=1}^{\ell_1} \pi(i_s M_i^1) \oplus \bigoplus_{i=1}^{\ell_2} \pi(i_s M_i^2) = M_{\ell_1}^1 10^* \oplus M_{\ell_2}^2 10^*\,. \tag{8.20}$$

Since $M^1 \neq M^2$ there are two cases:

1. $\ell_1 = \ell_2$, $M_{\ell_1}^1 10^* \neq M_{\ell_2}^2 10^*$, and $M_i^1 = M_i^2$ for all i, or
2. either $\ell_1 \neq \ell_2$ or there exists an i such that $M_i^1 \neq M_i^2$.

In the first case there is no collision, hence we focus on the second case. Without loss of generality we can assume that $M_i^1 \neq M_i^2$ for all i, and we can simplify the problem to calculating the probability that

$$\bigoplus_{i=1}^{\ell} \pi(x_i) = c\,, \tag{8.21}$$

where $\ell = \ell_1 + \ell_2$, $c = M_{\ell_1}^1 10^* \oplus M_{\ell_2}^2 10^*$, and $x_i \neq x_j$ for $i \neq j$.

Let $N = 2^n$, then $\mathbf{P}\left[\bigoplus_{i=1}^{\ell} \pi(x_i) = c\right]$ equals

$$\frac{1}{N!}\left|\left\{ y_1, \ldots, y_N \;\middle|\; \bigoplus_{i=1}^{\ell} y_i = c \text{ and } y_i \neq y_j \text{ for } i \neq j \right\}\right|. \tag{8.22}$$

By Lemma 3 we have that the probability is bounded above by $1/(N-\ell+1)$, giving us our desired result. □

Lemma 3. *Let $c \in \{0,1\}^n$ and let $N = 2^n$. The number of sequences $(y_1, y_2, \ldots, y_N) \in (\{0,1\}^n)^N$ with $y_i \neq y_j$ for $i \neq j$ such that*

$$\bigoplus_{i=1}^{\ell} y_i = c\,, \tag{8.23}$$

is not greater than $N!/(N-\ell+1)$.

Proof. We start by fixing y_1, for which there are N possibilities. Since y_2 cannot equal y_1, there are $N-1$ possibilities for y_2. Continuing this way, we have that there are $N-i$ possibilities for y_{i+1}, with $i \leq \ell - 2$. For y_ℓ there is at most one possibility, namely $c \oplus y_1 \oplus y_2 \oplus \cdots y_{\ell-1}$. All y_j for $j > \ell$ must be distinct from all preceding y_i, hence in total there are at most

$$N \cdot (N-1) \cdot \cdots \cdot (N-\ell+2) \cdot (N-\ell)! = \frac{N!}{N-\ell+1} \tag{8.24}$$

possible sequences. □

8.4 PMAC's Message Length Dependence

In contrast with CBC-MAC, EMAC, and LightMAC, the PMAC construction [47] stands out as having received little analysis showing the necessity of ℓ in the bound. It follows the protected counter sum design, and replaces PRF calls with tweakable block cipher calls. As a result, one would expect PMAC's security bound to be independent of the message length, which it is, if security is reduced to the PRP security of the tweakable block cipher. However, PMAC's tweakable block cipher is instantiated with an XE construction (Example 5.2.4), and the XE construction has a PRP advantage of $4.5q^2/2^n$, meaning a reduction to the PRP of the underlying tweakable block cipher would result in a quadratic message length dependence when using the XE construction. Nevertheless, Minematsu and Matsushima [128] were able to show that PMAC's security bound can be sharpened to $\ell q^2/2^n$, showing that PMAC's message length dependence is in the worst case linear.

No attacks are known which establish the linear dependence on message length in PMAC's security bound, hence it is not clear whether Minematsu and Matsushima's bound can be improved further. Furthermore, PMAC's basic structure lends itself to high-security extensions, such as PMAC-Plus [179], PMAC-with-Parity [180], and PMACX [185], where the latter two are designs which specifically minimize message length dependence as displayed in Figure 8.1 and Table 8.1.

In this section we study PMAC's message length dependence. We start by abstracting away details of PMAC in order to focus on its basic structure. We do so by considering *generic* PMAC, which is a generalized version of PMAC accepting an arbitrary block cipher and constants, and with an additional independent key. We prove that one of the following two statements is true:

1. either there are infinitely many instances of generic PMAC for which there are no attacks with success probability greater than $2q^2/2^n$,
2. or finding an attack against generic PMAC with success probability greater than $2q^2/2^n$ is computationally hard.

The second statement relies on a conjecture which we explain below.

Then we focus on an instantiation of generic PMAC, namely PMAC with Gray codes, introduced by Black and Rogaway [47]. We show that PMAC with Gray codes is an instantiation which does not meet the optimal bound of $2q^2/2^n$, by finding an attack with success probability $(2^{k-1}-1)/2^n$ with $\ell = 2^k$, establishing a dependence on ℓ for every power of two.

Approach. Proving the above results requires viewing the inputs to PMAC's block cipher calls in a novel way: as a set of points P lying in a finite affine plane. Keys are identified as slopes of lines in the affine plane. A collision is guaranteed to occur under a specific key w if and only if each line with slope w covers an even number of points in P; in this case we say that w *evenly covers* P.

Maximizing the collision probability means finding a set of points P for which there is large set of slopes W evenly covering P. But finding such a set W is non-trivial: the x-coordinates of the points in P must either contain a subset summing to zero, or satisfying some quadratic form.

Finding a subset summing to zero is the *subset sum* (SS) problem, which is known to be **NP**-complete. The second problem we call the *binary quadratic form* (BQF) problem (see Definition 8.4.8), and there is reason to believe this problem is **NP**-complete as well (see Appendix C, which contains a proof by Alan Szepieniec). As a result, we conjecture that finding solutions to the union of the two problems is computationally hard.

By reducing SS and the BQF problem to finding slopes W evenly covering points P, we establish our results.

Notation. If X is a set then $\overline{\mathsf{X}}$ is its complement. For this section, elements of X^q are denoted $\vec{x}$, with coordinates $(x_1, x_2, \ldots, x_q)$. If $f : \mathsf{X} \to \mathsf{Y}$ then define $\widetilde{f} : \mathsf{X}^+ \to \mathsf{Y}^+$ to be the mapping

$$\widetilde{f}(x_1, \ldots, x_q) = (f(x_1), \ldots, f(x_q)) \ . \tag{8.25}$$

If $\vec{a} \in \mathsf{X}^\ell$ and $\mu \leq \ell$, then $\vec{a}_{\leq \mu} \stackrel{\text{def}}{=} (a_1, a_2, \ldots, a_\mu)$. If X is a field, then for $\vec{a} \in \mathsf{X}^\ell$, $\vec{1} \cdot \vec{a} = \sum_{i=1}^{\ell} a_i$. Furthermore, when considering elements (x, y) of X^2, we call the left coordinate of the pair the x-coordinate, and the other the y-coordinate.

8.4.1 PMAC

PMAC is a VIL-PRF-based MAC, which means we can focus on the underlying VIL-PRF. Throughout this section we identify PMAC with its VIL-PRF. Furthermore, we focus on PMAC defined with a URP rather than a block cipher.

The original PMAC specifications [47, 153] have as message space the set of arbitrary length strings. Since our results focus on the dependency of PMAC on message length, it suffices to consider strings with length a multiple of some block size in order to illustrate how the security bounds evolve as a function of message length. With this in mind, we define PHASH, first introduced by Minematsu and Matsushima [128]. Figure 8.3 depicts a diagram of PHASH.

Definition 8.4.1 (PHASH)**.** Let X be a finite field of characteristic two with N elements. Let $\mathsf{M} \stackrel{\text{def}}{=} \mathsf{X}^{\leq N}$ and let $\vec{c} \in \mathsf{X}^N$ be a sequence containing all elements of X. Let π be a URP over X. Let $\omega = \pi(0)$, then PHASH : $\mathsf{M} \to \mathsf{X}$ is defined to be

$$\text{PHASH}(\vec{m}) \stackrel{\text{def}}{=} \vec{1} \cdot \widetilde{\pi}\left(\vec{m} + \omega \vec{c}_{\leq \ell}\right) \ , \tag{8.26}$$

where $\vec{m}$ has length ℓ.

PHASH maps messages to a single block. PMAC sends this block through a last transformation, whose output will be the tag. We describe two different generic versions of PMAC, one in which the last transformation is independent of PHASH, and one in which it is not.

Definition 8.4.2 (PMAC)**.** Consider PHASH : $\mathsf{M} \to \mathsf{X}$ with URP π and let c^* denote the last element of $\vec{c}$. If y is the output of PHASH under message $\vec{m}$, PMAC evaluated on $\vec{m}$ is $\pi(y + c^*\omega)$.

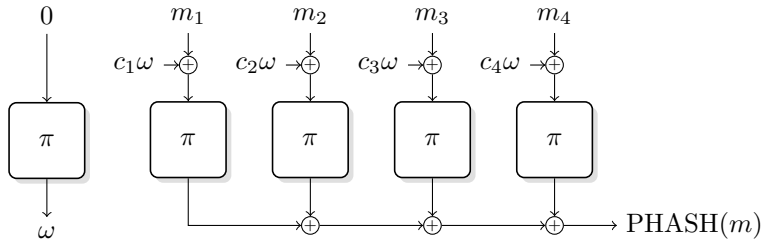


Figure 8.3: PHASH evaluated on a message $m = (m_1, m_2, m_3, m_4)$.

Definition 8.4.3 (PMAC*)**.** Consider PHASH : $\mathsf{M} \to \mathsf{X}$ with URP π. Let $\phi : \mathsf{X} \to \mathsf{X}$ be an independent URF. Then PMAC* is the composition of PHASH with ϕ.

Although PMAC* is defined with an independent outer URF instead of a URP, all the results in the section hold with slight modifications to the bounds if a URP is used.

The two specifications of PMAC define the sequence $\vec{c}$ differently. Our attack against PMAC applies to the specification with Gray codes [47], which we will define in Section 8.4.4. As pointed out by Nandi and Mandal [138], in order to get a PRF-advantage upper bound of the form $q^2\ell/N$, the only requirement on $\vec{c}$ is that each of its components are distinct.

8.4.2 PHASH Collision Probability

Definition 8.4.4. The collision probability of PHASH is

$$\max_{\vec{m}^1, \vec{m}^2 \in \mathsf{M}, \vec{m}^1 \neq \vec{m}^2} \mathbf{P}\left[\text{PHASH}(\vec{m}^1) = \text{PHASH}(\vec{m}^2)\right] . \tag{8.27}$$

PHASH's collision probability is closely linked with the security of PMAC and PMAC*. In particular, if an adversary finds a collision in PHASH, then it is able to distinguish PMAC and PMAC* from a URF. The converse is true for PMAC*, which is a well-known result; see for example Dodis and Pietrzak [70]. Concluding that a distinguishing attack against PMAC results in a collision found for PHASH has not been proven and is outside of the scope of the thesis, although we conjecture that the statement holds. In either case, understanding the effect of the message length on PHASH's collision probability will give us a good understanding of PMAC's message length dependence.

In this section we compute bounds on the collision probability for PHASH. Minematsu and Matsushima [128] prove an upper bound for the collision probability of PHASH. We use their proof techniques and provide a lower bound as well.

Throughout this section we fix two different messages $\vec{m}^1$ and $\vec{m}^2$ in M of length ℓ_1 and ℓ_2, respectively, and consider the collision probability over these messages. Let $\vec{m} = \vec{m}^1 \| \vec{m}^2$ and $\vec{d} = \vec{c}_{\leq \ell_1} \| \vec{c}_{\leq \ell_2}$.

If there exists i such that $m_i^1 = m_i^2$, then these blocks will cancel each other out in equation (8.27) and will not affect the collision probability, hence we remove them. Let $i_1, i_2, \ldots, i_k$ denote the indices of the blocks for which $\vec{m}^1$ equals $\vec{m}^2$, then define $\vec{m}^*$ to be $\vec{m}$ with the entries indexed by $i_1, i_2, \ldots, i_k$ and $i_1 + \ell_1, i_2 + \ell_1, \ldots, i_k + \ell_1$ removed; $\vec{d}^*$ is defined similarly and ℓ^* denotes the length of $\vec{m}^*$ and $\vec{d}^*$.

Let $\vec{x}^w \stackrel{\text{def}}{=} \vec{m}^* + w\vec{d}^*$ for $w \in \mathsf{X}$. The vector $\vec{x}^w$ represents the inputs to the permutation π when $\pi(0)$ equals w, meaning the equality $\text{PHASH}(\vec{m}^1) = \text{PHASH}(\vec{m}^2)$ can be written as

$$\vec{1} \cdot \widetilde{\pi}\,(\vec{x}^w) = 0\,, \tag{8.28}$$

given that $\pi(0) = w$. If there is a component of $\vec{x}^w$ which does not equal any of the other components, then equation (8.28) will contain a π-output which is roughly independent of the other outputs, thereby making a collision unlikely when $\pi(0) = w$. For example, say that $\vec{x}^w = (a, b, c, b)$, then equation (8.28) becomes $\pi(a) + \pi(b) + \pi(c) + \pi(b) = \pi(a) + \pi(c)$, which equals 0 with negligible probability.

Similarly, if there are an odd number of components of $\vec{x}^w$ which equal each other, but do not equal any other components, then they will not cancel out, resulting again in an unlikely collision. For example, if $\vec{x}^w = (a, a, a, b, b)$, then equation (8.28) becomes $\pi(a)$. In fact, a collision is only guaranteed under a given key w when each component of $\vec{x}^w$ is paired with another component so that each pair cancels each other out in equation (8.28). Bounding the collision probability in equation (8.27) amounts to determining how many keys w there are for which each component of $\vec{x}^w$ is paired.

We formalize these "equality classes" of components of $\vec{x}^w$ as follows. Define I to be the set of integers from 1 to ℓ^*, $\{1, \ldots, \ell^*\}$, then the components of $\vec{x}^w = (x_1^w, x_2^w, \ldots, x_{\ell^*}^w)$, induce the following equivalence relation on I: i is equivalent to j if and only if $x_i^w = x_j^w$. For $i \in I$, let $[i]$ denote i's equivalence class, and $\#[i]$ the number of elements in $[i]$. Let R^w denote the set of equivalence class representatives where each representative is the smallest element of its class. Let R_e^w be those $i \in R^w$ such that $\#[i]$ is even, and R_o^w the complement

of R_e^w in R^w. Taking the example $\vec{x}^w = (c, c, c, b, b, b, b, a)$, then R^w would equal $\{1, 4, 8\}$ and R_e^w is $\{4\}$.

Define $\mathbf{W}$ to be the set of $w \in \mathsf{X}$ such that R_o^w is empty. In other words, the set $\mathbf{W}$ is the set of keys w for which $\vec{m}^1$ and $\vec{m}^2$ are guaranteed to collide.

Proposition 8.4.1. *Let $F = PHASH$, then*

$$\frac{|\mathbf{W}|}{N} \leq \mathbf{P}\left[F(\vec{m}^1) = F(\vec{m}^2)\right] \leq \frac{|\mathbf{W}|}{N} + \frac{1}{N - \ell^* + 1} \ . \tag{8.29}$$

Proof. Let Π be the set of permutations on X. Let δ_w be the number of distinct components in $0\|\vec{x}^w$ and let S_w be the set of $\vec{y}$ such that $\vec{1} \cdot \vec{y} = 0$ and $w\|\vec{y}$ *matches* $0\|\vec{x}^w$, where two sequences $\vec{a}$ and $\vec{b}$ of the same length match if $a_i = a_j$ if and only if $b_i = b_j$, for all i, j. We have that

$$\mathbf{P}\left[F(\vec{m}^1) + F(\vec{m}^2) = 0\right] = \mathbf{P}\left[\vec{1} \cdot \widetilde{\pi}(\vec{x}^{\omega}) = 0\right] \tag{8.30}$$

$$= \frac{1}{N!} \cdot \left|\left\{p \in \Pi \;\middle|\; \vec{1} \cdot \widetilde{p}\left(\vec{x}^{p(0)}\right) = 0\right\}\right| \tag{8.31}$$

$$= \frac{1}{N!} \cdot \sum_{w \in \mathsf{X}} \sum_{\vec{y} \in S_w} \left|\left\{p \in \Pi \;\middle|\; \widetilde{p}(0\|\vec{x}^w) = w\|\vec{y}\right\}\right| \ . \tag{8.32}$$

Note that for all w and $\vec{y} \in S_w$,

$$\left|\left\{p \in \Pi \;\middle|\; \widetilde{p}(0\|\vec{x}^w) = w\|\vec{y}\right\}\right| = (N - \delta_w)! \ , \tag{8.33}$$

hence we get

$$\mathbf{P}\left[F(\vec{m}^1) = F(\vec{m}^2)\right] = \frac{1}{N!} \cdot \sum_{w \in \mathsf{X}} (N - \delta_w)! \cdot |S_w| \ . \tag{8.34}$$

Let $\vec{y}$ be such that $w\|\vec{y}$ matches $0\|\vec{x}^w$. Note that $y_i = y_j$ if and only if i is equivalent to j, and for any $i \in R^w$,

$$\sum_{j \in [i]} y_j = \begin{cases} 0 & \text{if } \#[i] \text{ is even} \\ y_i & \text{otherwise}\,. \end{cases} \tag{8.35}$$

Then $\vec{y} \in S_w$ if and only if $w\|\vec{y}$ matches $0\|\vec{x}^w$ and $\sum_{i \in R_o^w} y_i = 0$.

Let w be such that $x_i^w \neq 0$ for all i. The number of $\vec{y}$ such that $w\|\vec{y}$ matches $0\|\vec{x}^w$ and $\sum_{i \in R_o^w} y_i = 0$ can be counted as follows. Consider $\vec{y} = (y_1, \ldots, y_{\ell^*})$

satisfying the requirements, and enumerate the values in R_e^w: $i_1, i_2, \ldots, i_k$. By fixing $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$, we determine all components of $\vec{y}$ contained in the equivalence classes of R_e^w. Since $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$ is a sequence of k distinct values, all different from w, there are $(N-1)!/(N-k-1)!$ possibilities for $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$. If $R_o^w \neq \emptyset$, then we enumerate the elements of R_o^w: $j_1, j_2, \ldots, j_l$. Similar to R_e^w, by determining $y_{j_1}, y_{j_2}, \ldots, y_{j_l}$ we will determine the remaining components of $\vec{y}$. The sequence $y_{j_1}, y_{j_2}, \ldots, y_{j_l}$ contains l distinct values, all different from $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$ and w, and such that $y_{j_1} + y_{j_2} + \cdots + y_{j_l} = 0$, resulting in at most $(N-k-1)!/(N-k-l)!$ possibilities. Putting this together, and observing that $k+l = |R_e^w| + |R_o^w| = \delta_w - 1$, we get $|S_w| \leq \frac{(N-1)!}{(N-\delta_w+1)!}$ when $R_o^w \neq \emptyset$ and $x_i^w \neq 0$ for all i. If $R_o^w = \emptyset$, then $|S_w| = \frac{(N-1)!}{(N-\delta_w)!}$.

By following similar reasoning, we get that if w is such that there exists $x_i^w = 0$, $|S_w| \leq \frac{(N-1)!}{(N-\delta_w+1)!}$ when $R_o^w \neq \emptyset$, and $|S_w| = \frac{(N-1)!}{(N-\delta_w)!}$ otherwise.

Putting the above together, we have

$$\mathbf{P}\left[F(\vec{m}^1) = F(\vec{m}^2)\right] \leq \frac{|\mathbf{W}|}{N} + \frac{1}{N} \sum_{w \in \overline{\mathbf{W}}} \frac{1}{N - \delta_w + 1} \quad , \tag{8.36}$$

and since the computation of $|S_w|$ is exact when $R_o^w = \emptyset$, we get

$$\frac{|\mathbf{W}|}{N} \leq \mathbf{P}\left[F(\vec{m}^1) = F(\vec{m}^2)\right] \quad . \tag{8.37}$$

□

8.4.3 Necessary Conditions For a Collision

This section provides a geometric interpretation of the set $\mathbf{W}$ which facilitates finding necessary conditions for $\mathbf{W}$ to contain more than two elements.

Evenly Covered Sets. Recall that an element w of X is in $\mathbf{W}$ only if $R_o^w = \emptyset$, meaning $\#[i]$ is even for all $i \in R^w$. Two components x_i^w and x_j^w of $\vec{x}^w$ are equal if and only if

$$w = \frac{m_i^* - m_j^*}{d_j^* - d_i^*} \quad , \tag{8.38}$$

since the points such that $(d_i, m_i) = (d_j, m_j)$ were removed earlier when forming $\vec{m}^*$ from $\vec{m}$. In particular, equation (8.38) says that x_i^w equals x_j^w if and only if the points (d_i^*, m_i^*) and (d_j^*, m_j^*) lie on a line with slope w. Since $\#[i]$ is even, we know that there are an even number of points on the line through (d_i^*, m_i^*) with slope w, which motivates the following definition.

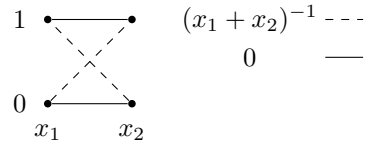


Figure 8.4: A set of four points evenly covered by the slopes 0 and $(x_1 + x_2)^{-1}$. The x-coordinates of the points are x_1 and x_2, and the y-coordinates are 0 and 1.

Definition 8.4.5. Let $\mathsf{P} \subset \mathsf{X}^2$ be a set of points. A line *evenly covers* P if it contains an even number of points from P. A slope $w \in \mathsf{X}$ evenly covers P if all lines with slope w evenly cover P. A subset of X evenly covers P if all slopes in the subset evenly cover P.

We let $\mathbf{P}$ denote the set of points (d_i, m_i) for $1 \leq i \leq \ell$. Applying the above definition together with equation (8.38), we get the following proposition.

Proposition 8.4.2. *An element $w \in \mathsf{X}$ is in $\mathbf{W}$ if and only if w evenly covers $\mathbf{P}$.*

Using this geometric interpretation, we obtain the upper bound proved by Minematsu and Matsushima [128] for the collision probability of PHASH.

Proposition 8.4.3.

$$|\mathbf{W}| \leq \ell^* - 1 \tag{8.39}$$

Proof. Given a point $p_0 \in \mathbf{P}$, all possible slopes connecting p_0 to another point in $\mathbf{P}$ can be generated from the lines connecting the points. This results in at most $|\mathbf{P}| - 1$ different slopes covering $\mathbf{P}$, hence an upper bound for $|\mathbf{W}|$ is $|\mathbf{P}| - 1 = \ell^* - 1$. □

It is easy to construct sets evenly covered by two slopes. Consider $\mathsf{P} \stackrel{\text{def}}{=} \{(x_1, 0), (x_1, 1), (x_2, 0), (x_2, 1)\}$, depicted in Figure 8.4. The possible slopes are 0 and $(x_1 + x_2)^{-1}$. Throughout the section we do not consider ∞ to be a slope, since such a slope would only be possible if $d_i^* = d_j^*$ in equation (8.38), which happens only if $m_i^* = m_j^*$. The lines with slope 0, from $(x_1, 0)$ to $(x_2, 0)$ and from $(x_1, 1)$ to $(x_2, 1)$, evenly cover P. Similarly, the lines with slope $(x_1+x_2)^{-1}$, from $(x_1, 0)$ to $(x_2, 1)$ and from $(x_1, 1)$ to $(x_2, 0)$, also evenly cover P. Therefore P is evenly covered by $\left\{0, (x_1 + x_2)^{-1}\right\}$.

The above set can be converted into two messages: $\vec{m}_1 = (0, 0)$ and $\vec{m}_2 = (1, 1)$. Setting $x_1 = c_1$ and $x_2 = c_2$, then we know that the collision probability of $\vec{m}_1$ and $\vec{m}_2$ is at least $2/N$.

Proposition 8.4.4. *There exist messages $\vec{m}_1$ and $\vec{m}_2$ such that $|\mathbf{W}| \geq 2$.*

Note that $\mathbf{P}$ constructed from $\vec{m}^*$ contains at most two points per x-coordinate.

Properties of Evenly Covered Sets. Although Proposition 8.4.3 gives a good upper bound for the collision probability of PHASH, it does not use any of the structure of evenly covered sets. In this section we explore various properties of evenly covered sets, allowing us to relate their discovery to **NP**-hard problems later.

The following lemma shows that removing an evenly covered subset from an evenly covered set results in an evenly covered set.

Lemma 4. *Let $\mathsf{P} \subset \mathsf{X}^2$ and let $\mathsf{W} \subset \mathsf{X}$ be a set evenly covering P. Say that P contains a subset P' evenly covered by W as well, then $\mathsf{P} \setminus \mathsf{P}'$ is evenly covered by W.*

Proof. Let $\mathsf{Q} \stackrel{\text{def}}{=} \mathsf{P} \setminus \mathsf{P}'$. The set W evenly covers Q if and only if every every line with slope $w \in \mathsf{W}$ contains an even number of points in Q. Let $p \in \mathsf{Q}$ and $w \in \mathsf{W}$ and consider the line λ with slope w through point p. By hypothesis, λ evenly covers P and P'. By removing P' from P, an even number of points are removed from λ, resulting in λ evenly covering Q. □

If a set P is evenly covered by at least two slopes u and v, then all the points in the set lie in a *loop*.

Definition 8.4.6. Let $\mathsf{P} \subset \mathsf{X}^2$ be evenly covered by $\mathsf{W} \subset \mathsf{X}$. A (u, v)-loop in (W, P) is a sequence of points $(p_1, p_2, \ldots, p_k)$ with two different slopes $u, v \in \mathsf{W}$ such that p_i and $p_{i+1 \pmod k}$ lie on a line with slope u for i odd, and on a line with slope v otherwise.

The set from Figure 8.4 contains $(0, (x_1 + x_2)^{-1})$-loops. In fact, there are always at least four points in any (u, v)-loop. Note that there must be at least three points since there are two distinct slopes. If there are only three points then p_1 is connected to p_2 via u, p_2 is connected to p_3 via v, and p_3 must be connected to p_1 via u, resulting in all three lying on the same line with slope u, but also p_2 lying on a line with slope v with p_3, resulting in a contradiction. Figure 8.5 shows a set with more complicated loops, including two which loop over all points in the set.

Lemma 5. *Let $\mathsf{P} \subset \mathsf{X}^2$ be evenly covered by $\mathsf{W} \subset \mathsf{X}$. Let $u, v \in \mathsf{W}$, then every point in P is in a (u, v)-loop starting with slope u and ending with slope v.*

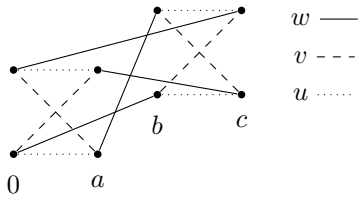


Figure 8.5: A set of points evenly covered by the slopes u, v, and w. Each point is accompanied by another point with the same x-coordinate. The x-coordinates of the pairs are indicated below the lower points.

Proof. Let $p_0 \in \mathsf{P}$, then by hypothesis there is another point p_1 in P lying on a line with slope u connecting to p_0. Similarly, there is a point p_2 different from p_0 and p_1 lying on a line with slope v connected to p_1. Continuing like this, we can create a sequence of points $p_0, p_1, \ldots, p_k$ until $p_{k+1} = p_i$ for some $i \leq k$, with the property that adjacent points in the sequence are connected by lines alternating with slope u and v.

If $i = 0$, then we are done. Otherwise, consider p_{i-1}, p_i, p_{i+1}, and p_k. Say that p_{i-1} is connected to p_i via a line with slope u, so that p_i is connected to p_{i+1} via a line with slope v. If p_k is connected to p_i via a line with slope v, then there are three points on the same line with slope v: p_i, p_{i+1}, and p_k. This means there is a fourth point p^* on the same line. Since p_k is connected to p_{i+1} via v, the sequence $p_{i+1}, p_{i+2}, \ldots, p_k$ forms a (u, v)-loop. We remove the (u, v)-loop from P, which is evenly covered by u and v, resulting in a set evenly covered by u and v, and we continue by induction. Similar reasoning can be applied when p_k is connected to p_i via u. □

Proposition 8.4.5. *The sum of the x-coordinates in a (u, v)-loop must be zero.*

Proof. Say that $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ are the points in the loop. Then

$$y_i + y_{i+1} = \delta_i (x_i + x_{i+1 \,(\mathrm{mod}\ k)}) \,, \tag{8.40}$$

where δ_i is u if i is odd, and v otherwise. Since

$$(y_1 + y_2) + (y_2 + y_3) + \cdots + (y_{k-1} + y_k) + (y_k + y_1) = 0 \,, \tag{8.41}$$

we have that

$$u(x_1 + x_2) + v(x_2 + x_3) + u(x_3 + x_4) + \cdots \\ + u(x_{k-1} + x_k) + v(x_k + x_1) = 0 \,, \tag{8.42}$$

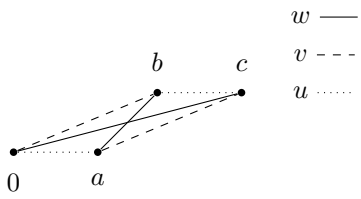


Figure 8.6: A set of points evenly covered by the slopes u, v, and w. None of the points are accompanied by another point with the same x-coordinate. The points are labelled by their x-coordinates.

therefore

$$(u+v)(x_1+x_2+\cdots+x_k)=0\,. \tag{8.43}$$

Since $u \neq v$, it must be the case that $x_1+x_2+\cdots+x_k=0$. □

Adversaries can only construct sets P where there are at most two points per x-coordinate. Therefore, either all loops only contain points (x,y) for which there is exactly one other point (x,y') with the same x-coordinate, or there exists a loop with a point which is the only one with that x-coordinate. For example, Figure 8.4 and Figure 8.5 depict evenly covered sets where every loop always contains all x-coordinate pairs. If we consider the only loop in Figure 8.4, then we get

$$0\cdot(x_1+x_2)+(x_1+x_2)^{-1}(x_2+x_1)+0\cdot(x_1+x_2)+(x_1+x_2)^{-1}(x_2+x_1)\,, \tag{8.44}$$

which trivially equals zero. All loops in Figure 8.5 also trivially sum to zero. In contrast, Figure 8.6 depicts an evenly covered set in which we get a non-trivial sum of the x-coordinates:

$$u\cdot a+v(a+c)+u(c+b)+v\cdot b=(u+v)(a+b+c)=0\,, \tag{8.45}$$

hence such a set only exists if $a+b+c=0$.

Therefore, Proposition 8.4.5 only poses a non-trivial restriction on the x-coordinates if there is a loop which contains a point without another point sharing its x-coordinate. If the loop contains all pairs of points with the same x-coordinates, then the x-coordinates will trivially sum to zero. This is why in the case of Figure 8.4 there are no restrictions on the x-coordinates, other than the fact that they must be distinct, resulting in the existence of sets evenly covered by two slopes.

In the case of Figure 8.5 however, there are additional restrictions on the x-coordinates. Consider the two points at x-coordinate 0. Then there is part

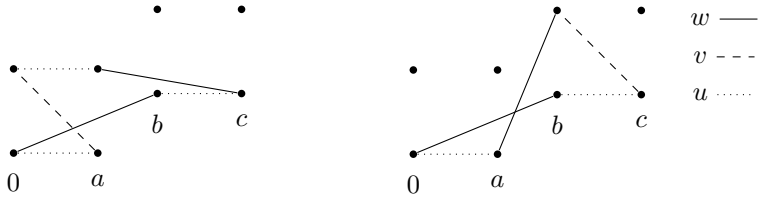


Figure 8.7: Illustration of loops with three slopes.

of a (u, v)-loop connecting them, and part of a (u, w)-loop connecting them, and combining both parts we get a full loop using all three slopes; see the left hand side of Figure 8.7. A similar loop involving all three slopes can be constructed around the points with x-coordinate b. Using these two loops, we get the following equations. From the left hand side of Figure 8.7 we have

$$ua + va = wb + u(b + c) + w(a + c) + ua \tag{8.46}$$

$$(u + v)a = (w + u)(a + b + c)\,. \tag{8.47}$$

From the right hand side of Figure 8.7 we have

$$(u + v)(b + c) = wb + ua + w(a + b) \tag{8.48}$$

$$(u + v)(b + c) = (w + u)a\,. \tag{8.49}$$

Combining both, we get the following:

$$\frac{a + b + c}{a} = \frac{a}{b + c} \tag{8.50}$$

$$a^2 + b^2 + c^2 + ab + ac = 0\,. \tag{8.51}$$

The last equation above can be described as a so-called *quadratic form*. A quadratic form over X is a homogeneous multivariate polynomial of degree two. In our case, the quadratic form can be written as $\vec{x}^T Q \vec{x}$, where $\vec{x} \in \mathsf{X}^n$ is the list of variables, and $Q \in \{0, 1\}^{n \times n}$ is a matrix with entries in $\{0, 1\}$. We say that $\vec{x}_*$ is a *solution* to Q if $\vec{x}_*^T Q \vec{x}_* = 0$, and the quadratic form Q is *non-trivial* if there exists $\vec{x} \neq 0$ such that $\vec{x}^T Q \vec{x} \neq 0$.

So the evenly covered set from Figure 8.5 only exists if the x-coordinates satisfy some non-trivial quadratic form. The same is true for any evenly covered set where all loops always contain pairs of points with the same x-coordinate.

Proposition 8.4.6. *Let $\mathsf{P} \subset \mathsf{X}^2$ be evenly covered by $\mathsf{W} \subset \mathsf{X}$ with $\mathsf{W} \geq 3$. Say that all loops in P contain only pairs of points with the same x-coordinates.*

Then there exists a subset S of k x-coordinates, and a non-trivial quadratic form described by a matrix $Q \in \{0,1\}^{k\times k}$ over k variables, such that when the k elements of S are placed in a vector $\vec{x}_* \in \mathsf{X}^k$, $\vec{x}_*^T Q \vec{x}_* = 0$.

Proof. Pick three slopes, u, v, w in W. We know that there are at least four points in P. Pick two pairs of points with the same x-coordinates: (p, p') and (q, q'). Consider the (u, v)-loop starting at p. By hypothesis it must contain p'. We let $\vec{a} = (a_1, a_2, \ldots, a_{k_a})$ denote the sequence of x-coordinates of the part of the (u, v)-loop from p to p'. Note that a_1 equals a_{k_a} since p and p' have the same x-coordinates. Similarly, the (u, v)-loop starting at q must contain q', and we denote the sequence of x-coordinates of the part of the (u, v)-loop from q to q' by $\vec{b} = (b_1, b_2, \ldots, b_{k_b})$. The same holds for the (v, w)-loops containing p and q, and we define the x-coordinate sequences $\vec{e}$ and $\vec{f}$ similarly.

Let y denote the difference in the y-coordinates of p and p'. For $\vec{a}$ we have the following:

$$u(a_1 + a_2) + v(a_2 + a_3) + \cdots + \delta(u, v)_{k_a}(a_{k_a-1} + a_{k_a}) = y\,, \tag{8.52}$$

where $\delta(u, v)_{k_a}$ is u if k_a is even and v otherwise. Collecting the terms, if k_a is even, we get

$$u(a_1 + a_2 + \cdots + a_{k_a-1} + a_{k_a}) + v(a_2 + \cdots + a_{k_a-1}) = y\,, \tag{8.53}$$

and since $a_1 = a_{k_a}$, we know that

$$(u + v)(a_2 + \cdots + a_{k_a-1}) = y\,. \tag{8.54}$$

If k_a is odd, then we get

$$(u + v)(a_1 + a_2 + \cdots + a_{k_a-1}) = y\,. \tag{8.55}$$

Note that it cannot be the case that $\sum a_i = 0$, since $y \neq 0$.

Similar reasoning applied to $\vec{b}$ gives

$$\begin{array}{ll} (v + w)(b_2 + \cdots + b_{k_b-1}) = y & \text{if } k_b \text{ is even} \\ (v + w)(b_1 + \cdots + b_{k_b-1}) = y & \text{otherwise}\,. \end{array} \tag{8.56}$$

Regardless of k_a and k_b's parities, setting both equations equal to each other results in the following equation:

$$\frac{u + v}{v + w} = \frac{\sum b_i}{\sum a_i}\,. \tag{8.57}$$

Applying the same result to $\vec{e}$ and $\vec{f}$, we get

$$\frac{u + v}{v + w} = \frac{\sum f_i}{\sum e_i}\,. \tag{8.58}$$

As a result, we have

$$\left(\sum b_i\right)\left(\sum e_i\right) + \left(\sum a_i\right)\left(\sum f_i\right) = 0\,, \tag{8.59}$$

which is the solution to a quadratic form. □

Computational Hardness As shown in Proposition 8.4.5 and Proposition 8.4.6, either there is a loop where the x-coordinates non-trivially sum to zero, or there is a subset of the x-coordinates which form the solution to some non-trivial quadratic form. The former is Subset Sum (SS), whereas the latter we name the binary quadratic form (BQF) problem.

Definition 8.4.7 (Subset Sum Problem (SS))**.** Given a finite field X of characteristic two and a subset $S \subset \mathsf{X}$, determine whether there is a subset $S_0 \subset S$ such that $\sum_{x \in S_0} x = 0$.

Definition 8.4.8 (Binary Quadratic Form Problem (BQF))**.** Given a finite field X of characteristic two and a subset $S \subset \mathsf{X}$, determine whether there is a non-trivial quadratic form $Q \in \{0,1\}^{k \times k}$ with a solution $\vec{x}_*$ made up of distinct components from S.

SS is know to be **NP**-complete. In Appendix C it is shown that BQF-t, a generalization of BQF, is **NP**-complete as well; the proof is due to Alan Szepieniec. The problem of finding either a subset summing to zero or a non-trivial quadratic form we call the *SS-or-BQF* problem.

Conjecture 1. There do not exist polynomial time algorithms solving SS-or-BQF.

Definition 8.4.9 (PHASH Problem)**.** Given a finite field X of characteristic two and a sequence of masks $\vec{c}$, determine whether there is a collision in PHASH with probability greater than $2/N$, where $N = |\mathsf{X}|$.

Given a collision in PHASH one can easily find a solution to SS-or-BQF. The converse does not necessarily hold, which means SS-or-BQF cannot be reduced to the PHASH problem in general, although we can conclude the following.

Theorem 18. *One of the following two statements holds.*

1. *There are infinitely many input sizes for which the PHASH problem does not have a solution, but SS-or-BQF does.*
2. *For sufficiently large input sizes, SS-or-BQF can be reduced to the PHASH problem.*

Proof. Both the PHASH and SS-or-BQF problems are decision problems, so the output of the algorithms solving the problems is a yes or a no, indicating whether the problems have a solution or not. Note that the inputs to both problems are identical. The reductions consist of simply converting the input to one problem into the input of the other, and then directly using the output of the algorithm solving the problem.

We proved that a yes instance for PHASH becomes a yes instance for SS-or-BQF: if you have an instance of SS-or-BQF, then you can convert it into a PHASH problem, and if you are able to determine that PHASH has a collision with sufficient probability, then SS-or-BQF has a solution. Similarly, a no instance for SS-or-BQF means a no instance for PHASH.

The issue is when there exists a no instance for PHASH and a yes instance for SS-or-BQF for a particular input size. If there are finitely many input sizes for which there is a no instance for PHASH and a yes instance for SS-or-BQF simultaneously, then there exists an r such that for all input sizes greater than r a no instance for PHASH occurs if and only if a no instance for SS-or-BQF occurs, and a yes instance for PHASH occurs if and only if a yes instance for SS-or-BQF occurs. Therefore, an algorithm which receives a no instance for PHASH can say that the corresponding SS-or-BQF problem is a no instance, and similarly for the yes instances, which is our reduction. Otherwise there are infinitely many input sizes for which PHASH is a no instance, and SS-or-BQF is a yes instance. □

If statement 1 holds, then there are infinitely many candidates for an instantiation of PMAC* with security bound independent of the message length. If statement 2 holds, and we assume that SS-or-BQF is hard to solve, then finding a collision for generic PHASH is computationally hard.

8.4.4 Finding Evenly Covered Sets

The previous section focused on determining necessary conditions for the existence of evenly covered sets, illustrating the difficulty with which such sets are found. Nevertheless, finding evenly covered sets becomes feasible in certain situations. In this section we provide an alternative description of evenly covered sets in order to find sufficient conditions for their existence.

Distance Matrices Let $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ be an enumeration of the elements of $\mathsf{P} \subset \mathsf{X}^2$. If $w \in \mathsf{X}$ covers P evenly, then the line with equation $y = w(x - x_1) + y_1$ must meet P in an even number of points. In particular,

there must be an even number of x_i values for which $w(x_i - x_1) + y_1 = y_i$, or in other words, the vector

$$w \cdot (x_1 - x_1, x_2 - x_1, \ldots, x_n - x_1) \tag{8.60}$$

must equal

$$(y_1 - y_1, y_2 - y_1, \ldots, y_n - y_1) \tag{8.61}$$

in an even number of coordinates. The same must hold for the lines starting from all other points in P.

Let $\Delta^{\vec{x}}$ be the matrix with (i, j) entry equal to $x_i - x_j$ and $\Delta^{\vec{y}}$ the matrix with (i, j) entry equal to $y_i - y_j$. We write $A \sim B$ if matrix $A \in \mathsf{X}^{n \times n}$ equals matrix $B \in \mathsf{X}^{n \times n}$ in an even number of entries in each row. Then, following the reasoning from above, we have that $w \in \mathsf{X}$ covers P evenly only if $\Delta^{\vec{y}} \sim w\Delta^{\vec{x}}$.

The matrices $\Delta^{\vec{x}}$ and $\Delta^{\vec{y}}$ are so-called *distance* matrices, that is, symmetric matrices with zero diagonal. Entry (i, j) in these distance matrices represents the "distance" between x_i and x_j, or y_i and y_j. In fact, starting from distance matrices M and D such that $M \sim wD$ we can also recover a set P evenly covered by w: interpret the matrices M and D as the distances between the points in the set P. This proves the following lemma.

Lemma 6. *Let $k \leq n - 1$ and let $\mathsf{W} \subset \mathsf{X}$ be a set of size k. There exist n by n distance matrices M and D such that $M \sim wD$ for all $w \in \mathsf{W}$ if and only if there exists P with $|\mathsf{P}| = n$ and W evenly covers P.*

From the above lemma we can conclude that the existence of $\mathsf{P} \subset \mathsf{X}^2$ evenly covered by $\mathsf{W} \subset \mathsf{X}$ is not affected by the following transformations:

1. Translating the set P by any vector in X^2. This also preserves the set W.
2. Subtracting any element $w_0 \in \mathsf{W}$ from the set W.
3. Scaling the set P in either x or y-direction by a non-zero scalar in X.
4. Scaling the set W by any non-zero element of X.

Connection with Graphs Let $\mathsf{P} \subset \mathsf{X}^2$ be evenly covered by $\mathsf{W} \subset \mathsf{P}$. The pair (P, W) has a natural graph structure with vertices P and an edge connecting two vertices p_1 and p_2 if and only if the line connecting them has slope in W. Figure 8.4 and Figure 8.5 provide diagrams which can also be viewed as examples of the natural graph structure. In this section we connect the existence of evenly covered sets with so-called *factorizations* of a graph. See Appendix B for a review of the basic graph theoretic definitions used in this section.

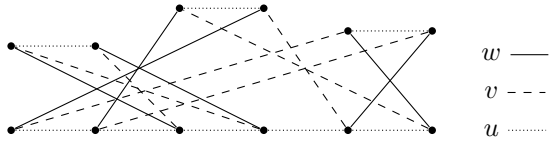


Figure 8.8: Non-trivial example of a set with 12 points evenly covered by three slopes. Horizontal points lie on the same y-coordinate, and vertical points on the same x-coordinate. Since there are six points on a line with slope u, the natural graph is not regular.

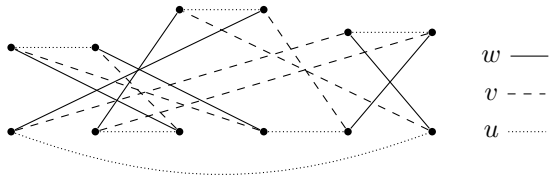


Figure 8.9: The diagram from Figure 8.8 converted into an associated graph. The slopes u, v, and w induce a natural 1-factorization of the graph.

Each vertex in the natural graph has at least $|\mathsf{W}|$ neighbours, and if there are two points per line in P, then the graph is $|\mathsf{W}|$-regular. Vertices have more than $|\mathsf{W}|$ neighbours only if they are on a line with more than two points. Since we are not interested in the redundancy from connecting a point with all points on the same line, we only consider graphs without the additional edges.

Definition 8.4.10. A graph associated to (P, W) is a $|\mathsf{W}|$-regular graph G with P as its set of vertices and an edge between two vertices p_1 and p_2 only if the line connecting p_1 with p_2 has slope in W.

Any graph associated to (P, W) is a subgraph of the natural graph structure described above, and there could be multiple associated graphs, depending upon what edges are chosen to connect multiple points lying on the same line. For example, Figure 8.8 depicts an evenly covered set with twelve points, six of which lie on the same line. As depicted in Figure 8.9, it can easily be converted into an associated graph.

The following definition allows us to describe another property that associated graphs have.

Definition 8.4.11. A k-factor of a graph G is a k-regular subgraph with the same vertex set as G. A k-factorization partitions the edges of a graph in disjoint k-factors.

Associated graphs have a 1-factorization induced by W, where each 1-factor is composed of the edges associated to the same slope in W. See Figure 8.9 for an example.

We know that every pair (P, W) has an associated $|\mathsf{W}|$-regular graph with 1-factorization. In order to determine the existence of evenly covered sets we need to consider when a k-regular graph with 1-factorization describes the structure of some pair (P, W) with $|\mathsf{W}| = k$. By first fixing a graph with a 1-factorization, it is possible to set up a system of equations to determine the existence of distance matrices M and D, and slopes W such that $M \sim wD$ for all $w \in \mathsf{W}$. Then, by applying Lemma 6, we will have our desired pair (P, W).

Definition 8.4.12. Let G be a regular graph with vertices $(v_1, \ldots, v_n)$ and a 1-factorization, and let $\mathsf{X}^{n\times n}$ denote the set of matrices over X. Define $\mathbf{S}_G \subset \mathsf{X}^{n\times n}$ to be the matrices where entry (i, j) equals entry (k, l) if and only if the edges (v_i, v_j) and (v_k, v_l) are in the same 1-factor of G.

Proposition 8.4.7. *There exists a set $\mathsf{P} \subset \mathsf{X}^2$ with n elements evenly covered by $\mathsf{W} \subset \mathsf{X}$ with $|\mathsf{W}| = k$ if and only if there exists a k-regular graph G of order n with a 1-factorization such that there is a solution to*

$$M = S \circ D\,, \tag{8.62}$$

where $S \in \mathbf{S}_G$, $M, D \in \mathsf{X}^{n\times n}$ are distance matrices, and $\circ$ denotes elementwise multiplication.

Therefore by picking a regular graph with a 1-factorization and solving a system of equations, we can determine the existence of pairs (P, W) for various sizes, in order to determine a lower bound for PHASH's collision probability.

Latin Squares and Abelian Subgroups In this section we consider what happens when we solve equation (8.62) with a 1-factorization of the complete graph of order n. Since we look at complete graphs, finding a solution would imply the existence of sets with n points evenly covered by $n - 1$ slopes, the optimal number as shown by Proposition 8.4.3. We describe a necessary and sufficient condition on the matrix D from equation (8.62), which in turn becomes a condition on the x-coordinates of the evenly covered sets.

As described by Laywine and Mullen [112, Sect. 7.3], 1-factorizations of a complete graph G of order n, with n even, are in one-to-one correspondence with reduced, symmetric, and unipotent Latin squares, that is, n by n matrices with entries in $\mathbb{N}$ such that

1. the first row enumerates the numbers from 1 to n,

1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2
8	7	6	5	4	3	2	1

Figure 8.10: A reduced, symmetric, unipotent Latin square of order eight corresponding to the Cayley table of the abelian 2-group of order eight.

2. the matrix is symmetric, that is, entry (i, j) equals entry (j, i),
3. the diagonal consists of just ones,
4. and each natural number from 1 to n appears just once in every row and column.

An example of such a Latin square can be found in Figure 8.10.

The correspondence between 1-factorizations of complete graphs and Latin squares works by identifying row i and column i with a vertex in the graph, labelling the 1-factor containing edge $(1, i)$ with i, and then setting entry (i, j) equal to the label of the 1-factor containing edge (i, j). This is exactly the structure of the matrices in $\mathbf{S}_G$.

Let n be a power of two. The *abelian 2-group of order n* is a commutative group in which every element has order two, that is, $a + a = 0$ for all elements a in the group. The Cayley table of the abelian 2-group of order n can be written as a reduced, symmetric, and unipotent Latin square. Fig. (8.10) provides an example of such a Cayley table, where 1 is identified with the identity of the group.

Definition 8.4.13. The (i, j) entry of the Cayley table of the abelian 2-group with ℓ elements is denoted $\gamma(i, j)$.

Lemma 7. $\gamma(i, \gamma(i, j)) = j$.

Proof. The Cayley table represents the operation of the abelian 2-group, where if $x + y = z$, then $x + z = y$. □

Proposition 8.4.8. *Let G denote the complete graph of order n, where n is a power of two, with 1-factorization induced by the Cayley table of the abelian 2-group of order n. Then Eq.* (8.62) *has a solution if and only if the first row of D forms an additive subgroup of X of order n.*

The above proposition shows that the graph structure corresponding to the abelian 2-group induces the same additive structure on the x-coordinates of the evenly covered set. This transfer of structure only works with this particular 1-factorization of the complete graph. In general, reduced, symmetric, and unipotent Latin squares do not even correspond to the Cayley table of some group: associativity is not guaranteed. Furthermore, 1-factorizations of non-complete graphs do not necessarily even form Latin squares; see for example Figure 8.8.

Proof. Denote the first row of S by $s_1, s_2, \ldots, s_n$, and the first row of D by $d_1, \ldots, d_n$. Note that D is entirely determined by its first row, since the (i, j) entry of D is $d_i + d_j$, and since S follows the form of γ, it is entirely determined by its first row as well. In particular, the (i, j) entry of S is $s_{\gamma(i,j)}$, where $\gamma(i, j)$ is the (i, j) entry of the Cayley table.

We need to determine the conditions under which $S \circ D$ is a distance matrix, as a function of $s_1, \ldots, s_n$ and $d_1, \ldots, d_n$. This happens if and only if the (i, j) entry of $S \circ D$ is equal to $s_i d_i + s_j d_j$:

$$s_i d_i + s_j d_j = s_{\gamma(i,j)}(d_i + d_j)\,. \tag{8.63}$$

Furthermore, it must be the case that

$$s_i d_i + s_{\gamma(i,j)} d_{\gamma(i,j)} = s_j(d_i + d_{\gamma(i,j)})\,, \tag{8.64}$$

since $\gamma(i, \gamma(i, j)) = j$. Therefore

$$s_j d_j + s_{\gamma(i,j)} d_{\gamma(i,j)} = s_{\gamma(i,j)}(d_i + d_j) + s_j(d_i + d_{\gamma(i,j)}) \tag{8.65}$$

$$(s_j + s_{\gamma(i,j)})(d_i + d_j + d_{\gamma(i,j)}) = 0\,. \tag{8.66}$$

Since S must follow the Latin square structure, the first row of S must consist of n distinct entries, hence $s_j \neq s_{\gamma(i,j)}$ and so $d_i + d_j + d_{\gamma(i,j)} = 0$. Therefore, $d_1, \ldots, d_n$ satisfies the equations of the Cayley table, hence they form an additive subgroup of X.

Continuing, we have the following equations:

$$s_i d_i + s_j d_j + s_{\gamma(i,j)} d_{\gamma(i,j)} = 0\,. \tag{8.67}$$

In order for these equations to be satisfied, $s_1 d_1, \ldots, s_n d_n$ must form an additive subgroup of X as well. In particular, there must exist an isomorphism ϕ mapping d_i to $s_i d_i$, which can be written as $d_i^{-1}\phi(d_i) = s_i$ for $i > 1$. The only requirement for the existence of such an isomorphism is that $x^{-1}\phi(x)$ must map to distinct values. Picking $x \mapsto x^2$ as the isomorphism, we have our desired result. Note that the d_i must be distinct, otherwise the s_i are not distinct, contradicting the fact that S follows the Latin square structure. □

Application to PMAC Before we present an attack, we first need the following lemma.

Lemma 8. *Let* P *and* P' *be disjoint subsets of* X^2 *evenly covered by* $\mathsf{W} \subset \mathsf{X}$. *Then* $\mathsf{P} \cup \mathsf{P}'$ *is evenly covered by* W.

Proof. Let λ be a line with slope $w \in \mathsf{W}$. Then λ contains an even number of points from P and an even number of points from P', and since P and P' are disjoint, λ contains an even number of points from $\mathsf{P} \cup \mathsf{P}'$. □

A collision in PHASH with probability $(\ell - 1)/N$ can be found as follows. Take $\vec{c}$ and let k be the smallest index such that $\vec{c}_{\leq k}$ contains a subsequence $\vec{c}'$ of length ℓ such that the elements $\{c'_1 + c'_1, c'_1 + c'_2, \ldots, c'_1 + c'_\ell\}$ form an additive subgroup of X. Let μ be the mapping which maps indices of $\vec{c}'$ onto indices of $\vec{c}$, so that $c'_i = c_{\mu(i)}$.

Let D be a distance matrix in $\mathsf{X}^{\ell \times \ell}$ such that its first row is equal to $(c'_1 + c'_1, c'_1 + c'_2, \ldots, c'_1 + c'_\ell)$; recall that a distance matrix is completely determined by its first row. Let G be the complete graph of order ℓ with 1-factorization determined by the abelian 2-group of order ℓ. Solve equation (8.62), that is, find a distance matrix M such that there exists $S \in \mathbf{S}_G$ where

$$M = S \circ D\,. \tag{8.68}$$

Let $\vec{m}^1$ denote the first row of M, and let W denote the elements making up the first row of S, without the first row element. Then the set $\mathsf{P} \stackrel{\text{def}}{=} \{(c'_1, m^1_1), \ldots, (c'_\ell, m^1_\ell)\}$ is evenly covered by W, which contains $\ell - 1$ slopes.

By translating P vertically by some constant, say 1, construct the disjoint set P', which is also evenly covered by W. Therefore, by Lemma 8, the union of P and P' is evenly covered by W. Let $\vec{m}^2$ denote the y-coordinates of P'.

Define $\overline{\vec{m}^1}$ to be the vector of length k where for all $i \leq \ell$, $\overline{m^1}_{\mu(i)} = m^1_i$, and for all i not in the range of μ, $\overline{m^1}_i = 0$. Define $\overline{\vec{m}^2}$ similarly. Then $\overline{\vec{m}^1}$ and $\overline{\vec{m}^2}$ collide with probability $(\ell - 1)/N$.

For sufficiently large k, $\vec{c}_{\leq k}$ will always contain additive subgroups. In particular, one can find such subgroups in PMAC with Gray codes [47], where $\vec{c}$ is defined as follows. In this case $\mathsf{X} \stackrel{\text{def}}{=} \{0,1\}^\nu$ is the set of ν-bit strings, identified in some way with a finite field of size 2^ν. We define the following sequence of vectors λ^ν:

$$\lambda^1 = (0, 1) \tag{8.69}$$

$$\lambda^{\nu+1} = (0\|\lambda^\nu_1, 0\|\lambda^\nu_2, \ldots, 0\|\lambda^\nu_{2^\nu}, 1\|\lambda^\nu_{2^\nu}, \ldots, 1\|\lambda^\nu_2, 1\|\lambda^\nu_1)\,. \tag{8.70}$$

Note that λ^ν contains all strings in X. Then $\vec{c}$ is λ^ν without the first component, meaning $\vec{c}$ contains all strings in X without the zero string. Similarly, the sequence $(c_1, \ldots, c_{2^\kappa})$ contains all strings starting with $\nu - \kappa$ zeros, i.e. $0^{\nu-\kappa} \| \{0,1\}^\kappa$, excluding the zero string. Note that $c_1 = 0^{\nu-1}1$. The sequence $(c_1 + c_1, c_1 + c_2, \ldots, c_1 + c_{2^\kappa})$ contains all strings in $0^{\nu-\kappa} \| \{0,1\}^\kappa$ except for c_1, meaning it contains an additive subgroup of order $2^{\kappa-1}$. This results in an attack using messages of length $k = 2^\kappa$ with success probability $(2^\kappa - 1)/2^\nu$.

Chapter 9

Conclusion

9.1 Review

In Chapter 3 we reviewed the basic concepts and definitions on achieving confidentiality and integrity. Encryption schemes were reviewed, which aim to provide confidentiality, and authenticators introduced, a definition focusing on the details necessary to achieve integrity. Authenticators describe both MAC algorithms and AE schemes. AE schemes were subsequently introduced in Chapter 3 as being the constructions which aim for both confidentiality and integrity.

In Chapter 4 we reviewed the IV-based extensions of all the definitions from Chapter 3. For integrity the IV-formalization does not make a difference, but we saw that confidentiality falls apart in the abused IV setting. We provided new definitions of abused IV confidentiality which align more closely to intuition, since they show that security is never achieved in the abused IV setting.

Chapter 5 covered all the necessary building blocks to construct encryption schemes, MAC algorithms, and AE schemes. In this chapter the tweakable online cipher variants of COPE and COBRA were introduced, and compared with the tweakable online cipher variant of TC3. Many of the examples in the chapter were given as modes of operation for tweakable block ciphers, even if they were introduced as a mode of operation for block ciphers.

Chapter 6 discussed how to achieve integrity and confidentiality in all the IV settings using the building blocks from Chapter 5. The issue of ciphertext expansion was discussed, along with a new application of ciphertext stealing to

COPE in order to preserve length. The many ways of adding an integrity check to an encryption scheme were discussed, including the OCB trick, which was applied to COPE in order to construct COPA.

Chapter 7 discussed the issues of how implementations of AE schemes in practice might not align with the assumptions made in theory. The Subtle AE framework was reviewed, which describes all possible forms of implementation leakage that could occur in practice. The releasing unverified plaintext definitions were then viewed as a special case of the subtle AE framework. The reasons for why many AE schemes do not achieve RUP security were discussed, and solutions were presented as well. For integrity in the RUP setting, the PRF-to-IV construction was reviewed, as well as the attack on OCB.

Finally, in Chapter 8 we discussed what the security loss in reductions means to practice. In the case of lightweight block ciphers, we saw that their small block sizes could impose impractical limits on how much data could be processed under a single key. To alleviate the problem, we introduced LightMAC, a simple MAC algorithm whose security bound does not degrade as a function of the message length. Then PMAC was analyzed, a known MAC algorithm. Its dependence on message length had not been explored before, and we showed how it depended on the masks used for PMAC's block cipher calls. If the masks are Gray codes, then we illustrated an attack establishing a dependence on message length.

9.2 Open Problems

Design. Both COBRA and POET were originally published with faulty security proofs and subsequently attacked [133], and COPA originally used the XLS construction to deal with ciphertext expansion, which was shown to be weak as well [135], resulting in a worse integrity bound for COPA [137]. Faulty proofs tend to have a detrimental effect on security, since the difference between a secure and an insecure scheme can be small, and often non-intuitive. Furthermore, increased design complexity and the push for greater efficiency means that proving the security of algorithms will not become simpler in the future.

Other than the issue of faulty proofs, the current design approach uses intuition and trial and error to search for optimally efficient schemes. However, the search space for secure and efficient schemes is large, and there is no reason to believe that human intuition will be able to find the best schemes in this large space.

One promising approach is to explore what the limits are of the search space:

how many block cipher calls must a tweakable cipher have in order to provide security, how efficient can the intermediate operations be, and is it possible to efficiently avoid ciphertext expansion? Although Nandi [136] has made some progress in this direction by considering the efficiency of encryption modes of operation with linear intermediate functions, little progress has been made in characterizing the entire search space for encryption modes, let alone any of the other building blocks.

An alternative is to automate the search for secure schemes, an approach taken by Hoang, Katz, and Malozemoff [91], who automate the search for secure AE modes of operation for tweakable block ciphers. They consider a restricted class of modes, but are able to discover interesting variants of known modes. Further automation might even obviate the need for proofs if the search is able to prune insecure schemes.

Subtle AE and RUP. The RUP setting seems to place strict limits on the efficiency of the schemes, since all known solutions use tweakable ciphers. Are there more efficient constructions? Alternatively, is there a way to meaningfully weaken the Λ-function so as to provide sufficient RUP-security with known constructions?

Message Length. LightMAC was introduced as a simple construction with an ℓ-free bound, and it performs favourably in comparison with other MACs providing ℓ-free bounds, namely PMAC-with-Parity [180] and PMACX [185]. However, the question of what the most efficient possible construction is remains open. Some instantiation of PMAC could be a contender, although it is unclear what PMAC's security bound looks like when other masks are used. In particular, the security of PMAC's other variant, with *powering up* masks [153], is still open, since it is not clear when they form an additive subgroup, nor is it clear what other sufficient conditions there are for finding evenly covered sets. Finally, Chapter 8 also shows beyond-birthday bound constructions like 3kf9 [183], PMAC_Plus [179], and the Sum of CBCs [178], which are able to process many more messages than the square root of the block size (but not very long messages). Note that they are easily identified in Figure 8.1 by the fact that their graphs do not go to zero on the right hand side of the figure. An obvious question is how to efficiently construct a beyond-birthday bound MAC algorithm which provides minimal dependence on the message length.

Appendix A

COBRA ciphertext stealing

Let M be a message where $M_1 M_2 \cdots M_{2\ell-1} M_{2\ell} = M$ and $|M_i| = n$ for $1 \leq i < 2\ell - 1$.

A.1 $\ell > 1$, $|M_{2\ell-1}| = n$, and $0 < |M_{2\ell}| < n$

We start by computing the ciphertext of $M_1 \cdots M_{2\ell-2}$ as is usually done in COBRA, resulting in $C_1 \cdots C_{2\ell-2}$. Let M^* denote the rightmost $|M_{2\ell}|$ bits of $C_{2\ell-2}$, and we write $C_{2\ell-2} = C'_{2\ell-2} M^*$. Then we compute the final ciphertext fragment $C_{2\ell-1} C_{2\ell}$ using $M_{2\ell-1} M_{2\ell} M^*$ as our "new" final message fragment, using different tweaks for the final block cipher calls. The resulting ciphertext is

$$C_1 \cdots C_{2\ell-3} C'_{2\ell-2} C_{2\ell-1} C_{2\ell}. \tag{A.1}$$

Figure A.1 shows a diagram of the process. Note that we can recover M^* with just knowledge of $C_{2\ell-1}$ and $C_{2\ell}$:

$$M_{2\ell} M^* = \left[C_{2\ell} \oplus \mathsf{E}_K^{(N,\ell,4)}(C_{2\ell-1}) \right] \oplus \left(\left[\mathsf{E}_K^{(N,\ell,3)}\left(C_{2\ell} \oplus \mathsf{E}_K^{(N,\ell,4)}(C_{2\ell-1}) \right) \oplus C_{2\ell-1} \right] \otimes L \right).$$

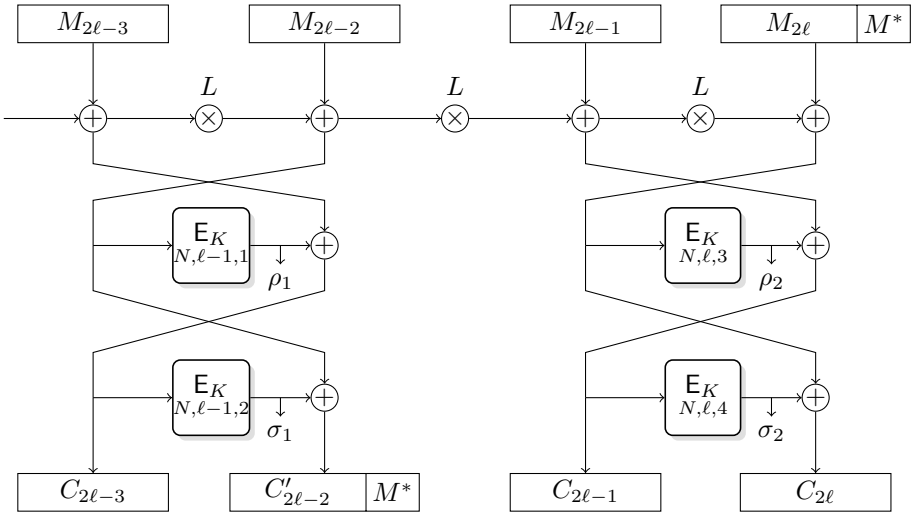


Figure A.1: Messages where the last block is not of full length, i.e. $0 < |M_{2\ell}| < n$. Here M^* is "stolen" from ciphertext block $C_{2\ell-2}$ and used in the input to the final fragment.

A.2 $\ell > 2$ and $0 < |M_{2\ell-1}| \leq n$

When there is no last block $M_{2\ell}$, we replace it with the preceding ciphertext block, $C_{2\ell-2}$. Then we steal ciphertext M^* of length $|M_{2\ell-1}|$ from the ciphertext block $C_{2\ell-4}$ such that $C_{2\ell-4} = C'_{2\ell-4}M^*$. The rest of the computation is similar to the previous case (Section A.1) and is depicted in Figure A.2.

A.3 $|M| \leq 3n$

The above methods only work for messages of length greater than $3n$ (otherwise there is no ciphertext to steal from). We need to use different techniques in order to deal with shortest messages.

For $2n < |M| \leq 3n$ we can use a technique similar as to what is used in COPA. Instead of using XLS [150] which uses the inverse block cipher and was shown to be insecure, we can use HCH [59] in order to compute the output as follows:

$$C_1C_2T' \leftarrow \mathcal{E}(M_1M_2) \tag{A.2}$$

$$C_3T \leftarrow \mathrm{HCH}(M_3T'), \tag{A.3}$$

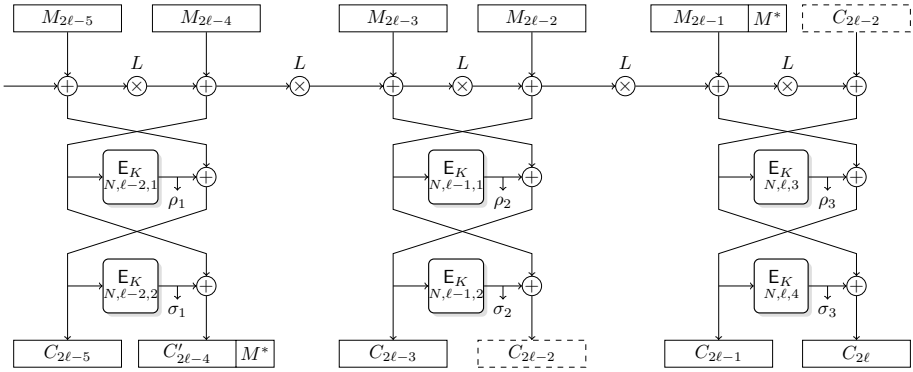


Figure A.2: Messages where the last fragment is of length less than or equal to n, i.e. $0 < |M_{2\ell-1}| \leq n$. Here M^* is stolen from ciphertext block $C_{2\ell-4}$ and used in the input to the final fragment together with ciphertext fragment $C_{2\ell-2}$.

where $\mathcal{E}$ denotes COBRA and the final output of the scheme is $C_1C_2C_3T$.

Appendix B

Basic Graph Theoretic Definitions

1. A *neighbour* of a vertex v in a graph G is a vertex with an edge connecting it to v.
2. A graph G is said to be *k-regular* if every vertex of G has exactly k neighbours.
3. A *subgraph* of a graph G is a graph with vertex set and edge set subsets of G's vertex and edge sets, respectively.
4. A *complete graph* is a graph in which every vertex is connected to every other vertex via an edge.

Appendix C

BQF-t is NP-complete

This appendix is due to Alan Szepieniec.

Definition C.0.1 (BQF-t)**.** Given a finite field X with characteristic 2 and a vector $\boldsymbol{x}_* \in \mathsf{X}^k$ and a target element $t \in \mathsf{X}$, determine if there is a non-trivial binary quadratic form $Q \in \{0,1\}^{k \times k}$ such that $\boldsymbol{x}_*^T Q \boldsymbol{x}_* = t$.

Note. The word 'binary' in our use of the term 'binary quadratic form' refers to the coefficients of the quadratic form matrix Q and not to the number of variables.

Proposition C.0.1. *BQF-t* $\in$ **NP**

Proof. Given a BQF-t yes-instance $(\mathsf{X}, \boldsymbol{x}_*, t)$ of $(k+2) \times \ell$ bits, there exists a certificate of $k^2 \times \ell$ bits that proves it is a yes-instance, namely the matrix Q such that $\boldsymbol{x}_*^T Q \boldsymbol{x}_* = t$. Moreover, the validity of this certificate can be verified by computing $\boldsymbol{x}_*^T Q \boldsymbol{x}_*$ and testing if it is indeed equal to t. This evaluation requires $(n+1) \times n$ multiplications and the same number of additions in the finite field X. After testing equality, the non-triviality of Q is verified by testing whether $Q^T + Q \neq 0$, costing another n^2 finite field additions and as many equality tests. Thus, for every yes-instance of BQF-t, there exists a polynomial-size certificate whose validity is verifiable in polynomial time. Hence, BQF-t $\in$ **NP**. □

Proposition C.0.2. *BQF-t is* **NP***-hard.*

Proof. We show that BQF-t is **NP**-hard by reducing the subset-sum problem SS, another **NP**-hard problem, to it. In particular, we show that SS $\leq$ BQF-t under deterministic polynomial-time Karp reductions.

Given an instance (X, S) of SS, the goal is to find a subset $S_0 \subset S$ such that $\sum_{x \in S_0} x = 1$. Note the target of SS can be changed without loss of generality. We transform this problem instance to an instance $(\mathsf{X}', \boldsymbol{x}_*, t)$ of BQF-t as follows.

Let $k = \#S$, the number of elements in S and let each unique element s_i of S be indexed by $i \in \{1, \ldots, k\}$. Choose a degree $2k+1$ irreducible polynomial $\psi(z) \in \mathsf{X}[z]$ and define the extension field $\mathsf{X}' = \mathsf{X}[z]/\langle\psi(z)\rangle$. Then define the vector $\boldsymbol{x}_*$ as follows:

$$\boldsymbol{x}_* = \begin{pmatrix} z^1 s_1 \\ z^2 s_2 \\ \vdots \\ z^k s_k \\ z^{-1} \\ z^{-2} \\ \vdots \\ z^{-k} \end{pmatrix} .$$

The BQF-t instance is $(\mathsf{X}', \boldsymbol{x}_*, 1)$. It now remains to be shown that 1) this transformation is computable in polynomial time; 2) if the SS problem instance is a yes-instance, then the BQF-t problem instance is yes-instance; 3) conversely, if the SS problem instance is a no-instance, then the BQF-t problem instance is a no-instance.

1. It is known to be possible to deterministically select an irreducible polynomial over a finite field of small characteristic in polynomial time [164]. After selecting the polynomials, the inverse of z is computed using the polynomial-time extended GCD algorithm and all the necessary powers of z and z^{-1} are found after two times k multiplications. Lastly, the proper powers of z are combined with the s_i elements using k multiplications for the construction of the first half of the vector $\boldsymbol{x}_*$; the second half of this vector has already been computed. So since this transformation consists of a polynomial-number of polynomial-time steps, its total running time is also polynomial.

2. If the SS instance is a yes-instance, then there exist k binary weights $w_i \in \{0, 1\}$ for all $i \in \{1, \ldots, k\}$ such that $\sum_{i=1}^{k} w_i s_i = 1$. The existence of these weights imply the existence of the matrix Q, as defined below. This matrix consists of four $k \times k$ submatrices and only the diagonal of

the upper right submatrix is nonzero. In fact, this diagonal is where the weights w_i appear.

$$Q = \left(\begin{array}{c|c} & w_1 \\ & \ddots \\ & w_k \\ \hline & \end{array}\right) \tag{C.1}$$

Indeed, the BQF-t instance is guaranteed to be a yes-instance as

$$\boldsymbol{x}_*^T Q \boldsymbol{x}_* = \sum_{i=1}^{k} z^i s_i w_i z^{-i} = 1$$

if and only if

$$\sum_{i=1}^{k} w_i s_i = 1 \ ,$$

which is the solution to the SS problem. Also, Q is non-trivial if there exists at least one nonzero weight w_i.

3. If the SS instance is a no-instance, then no set of weights w_i such that $\sum_{i=1}^{k} w_i s_i = 1$ exists. Consequently, no Q satisfying $\boldsymbol{x}_*^T Q \boldsymbol{x}_* = 1$ can exist. The reason is that all the elements of the Q-matrix except for the upper right diagonal are multiplied with higher or lower powers of z, which make them linearly independent from 1. Hence, neither the upper right diagonal nor any other set of nonzero elements in Q can make the total quadratic form equal to one.

□

Corollary 1. *BQF-t is* **NP***-complete.*

Bibliography

[1] The Alert attack. `https://www.mitls.org/pages/attacks/Alert`. Date accessed 2016.03.03.

[2] CWI cryptanalyst discovers new cryptographic attack variant in Flame spy malware. `http://www.cwi.nl/news/2012/cwi-cryptanalist-discovers-new-cryptographic-attack-variant-in-flame-spy-malware`, June 2012. Date accessed 2016.03.04.

[3] Abed, F., Fluhrer, S. R., Forler, C., List, E., Lucks, S., McGrew, D. A., and Wenzel, J. Pipelineable On-line Encryption. In Cid and Rechberger [61], pp. 205–223.

[4] Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Heninger, N., Springall, D., Thomé, E., Valenta, L., VanderSloot, B., Wustrow, E., Zanella-Béguelin, S., and Zimmermann, P. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *22nd ACM Conference on Computer and Communications Security* (Oct. 2015).

[5] Akdemir, K., Dixon, M., Feghali, W., Fay, P., Gopal, V., Guilford, J., Erdinc Ozturk, G. W., and Zohar, R. Breakthrough AES Performance with Intel AES New Instructions. Intel white paper, January 2010.

[6] Albrecht, M. R., Driessen, B., Kavun, E. B., Leander, G., Paar, C., and Yalçın, T. Block Ciphers - Focus on the Linear Layer (feat. PRIDE). In Garay and Gennaro [79], pp. 57–76.

[7] Albrecht, M. R., Paterson, K. G., and Watson, G. J. Plaintext Recovery Attacks against SSH. In *IEEE Symposium on Security and Privacy* (2009), IEEE Computer Society, pp. 16–26.

[8] AlFardan, N. J., and Paterson, K. G. Lucky Thirteen: Breaking the TLS and DTLS Record Protocols. In *IEEE Symposium on Security and Privacy* (2013), IEEE Computer Society, pp. 526–540.

[9] Anderson, E., Beaver, C. L., Draelos, T., Schroeppel, R., and Torgerson, M. ManTiCore: Encryption with Joint Cipher-State Authentication. In *ACISP* (2004), H. Wang, J. Pieprzyk, and V. Varadharajan, Eds., vol. 3108 of *Lecture Notes in Computer Science*, Springer, pp. 440–453.

[10] Andreeva, E., Barwell, G., Page, D., and Stam, M. Turning Online Ciphers Off. Cryptology ePrint Archive, Report 2015/485, 2015.

[11] Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., and Yasuda, K. APE: authenticated permutation-based encryption for lightweight cryptography. In Cid and Rechberger [61], pp. 168–186.

[12] Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., and Yasuda, K. How to Securely Release Unverified Plaintext in Authenticated Encryption. In Sarkar and Iwata [160], pp. 105–125.

[13] Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., and Yasuda, K. Parallelizable and Authenticated Online Ciphers. In Sako and Sarkar [159], pp. 424–443.

[14] Andreeva, E., Luykx, A., Mennink, B., and Yasuda, K. COBRA: A Parallelizable Authenticated Online Cipher Without Block Cipher Inverse. In *Fast Software Encryption, FSE 2014* (London,UK, 2014), C. Cid and C. Rechberger, Eds., Lecture Notes in Computer Science, Springer-Verlag, p. 16.

[15] Aoki, K., and Yasuda, K. The Security of the OCB Mode of Operation without the SPRP Assumption. In *ProvSec 2013* (2013), W. Susilo and R. Reyhanitabar, Eds., vol. 8209 of *Lecture Notes in Computer Science*, Springer, pp. 202–220.

[16] Arbaugh, W., Shankar, N., Wan, Y., and Zhang, K. Your 80211 wireless network has no clothes. *Wireless Communications, IEEE 9*, 6 (2002), 44–51.

[17] Atlantic, T. The Inside Story of How Facebook Responded to Tunisian Hacks. `http://www.theatlantic.com/technology/archive/2011/01/the-inside-story-of-how-facebook-responded-to-tunisian-hacks/70044/`, January 2011. Date accessed 2016.03.06.

[18] Aviram, N., Schinzel, S., Somorovsky, J., Heninger, N., Dankel, M., Steube, J., Valenta, L., Adrian, D., Halderman, J. A., Dukhovni, V., Käsper, E., Cohney, S., Engels, S., Paar, C., and Shavitt, Y. The DROWN Attack. https://drownattack.com/. Date accessed 2016.03.03.

[19] Bangeman, E. Blame for record-breaking credit card data theft laid at the feet of WEP. http://arstechnica.com/security/2007/05/blame-for-record-breaking-credit-card-data-theft-laid-at-the-feet-of-wep/, May 2007. Date accessed 2016.03.04.

[20] Barker, W. C., and Barker, E. *Recommendation for the triple data encryption algorithm (TDEA) block cipher*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2004.

[21] Barwell, G., Page, D., and Stam, M. Rogue Decryption Failures: Reconciling AE Robustness Notions. In *Cryptography and Coding - 15th IMA International Conference, IMACC 2015, Oxford, UK, December 15-17, 2015. Proceedings* (2015), J. Groth, Ed., vol. 9496 of *Lecture Notes in Computer Science*, Springer, pp. 94–111.

[22] Baysal, A., and Sahin, S. RoadRunneR: A Small And Fast Bitslice Block Cipher For Low Cost 8-bit Processors. LightSec 2015, 2015. to appear.

[23] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013.

[24] Bellare, M., Boldyreva, A., Knudsen, L. R., and Namprempre, C. Online Ciphers and the Hash-CBC Construction. In *CRYPTO* (2001), J. Kilian, Ed., vol. 2139 of *Lecture Notes in Computer Science*, Springer, pp. 292–309.

[25] Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. A Concrete Security Treatment of Symmetric Encryption. In *FOCS* (1997), IEEE Computer Society, pp. 394–403.

[26] Bellare, M., Guérin, R., and Rogaway, P. XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions. In Coppersmith [62], pp. 15–28.

[27] Bellare, M., Kilian, J., and Rogaway, P. The Security of Cipher Block Chaining. In *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA,*

August 21-25, 1994, Proceedings (1994), Y. Desmedt, Ed., vol. 839 of *Lecture Notes in Computer Science*, Springer, pp. 341–358.

[28] Bellare, M., Kilian, J., and Rogaway, P. The Security of the Cipher Block Chaining Message Authentication Code. *J. Comput. Syst. Sci. 61*, 3 (2000), 362–399.

[29] Bellare, M., Kohno, T., and Namprempre, C. Breaking and Provably Repairing the SSH Authenticated Encryption Scheme: A Case Study of the Encode-then-Encrypt-and-MAC Paradigm. *ACM Transactions on Information and System Security* (2004), 206–241.

[30] Bellare, M., Krovetz, T., and Rogaway, P. Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding* (1998), K. Nyberg, Ed., vol. 1403 of *Lecture Notes in Computer Science*, Springer, pp. 266–280.

[31] Bellare, M., and Micciancio, D. A New Paradigm for Collision-Free Hashing: Incrementality at Reduced Cost. In *EUROCRYPT* (1997), W. Fumy, Ed., vol. 1233 of *Lecture Notes in Computer Science*, Springer, pp. 163–192.

[32] Bellare, M., and Namprempre, C. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In *ASIACRYPT 2000* (2000), T. Okamoto, Ed., vol. 1976 of *Lecture Notes in Computer Science*, Springer, pp. 531–545.

[33] Bellare, M., Pietrzak, K., and Rogaway, P. Improved Security Analyses for CBC MACs. In *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings* (2005), V. Shoup, Ed., vol. 3621 of *Lecture Notes in Computer Science*, Springer, pp. 527–545.

[34] Bellare, M., and Rogaway, P. Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In *ASIACRYPT* (2000), T. Okamoto, Ed., vol. 1976 of *Lecture Notes in Computer Science*, Springer, pp. 317–330.

[35] Bellare, M., and Rogaway, P. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings* (2006),

S. Vaudenay, Ed., vol. 4004 of *Lecture Notes in Computer Science*, Springer, pp. 409–426.

[36] Berendschot, A., Boly, J.-P., Bosselaers, A., Brandt, J., Chaum, D., Damgård, I., de Rooij, P., Dichtl, M., Fumy, W., Jansen, C. J. A., Landrock, P., Preneel, B., Roelofsen, G., van der Ham, M., and Vandewalle, J. *Integrity Primitives for Secure Information systems. Final Report of RACE Integrity Primitives Evaluation (RIPE-RACE 1040)*, vol. 1007 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.

[37] Bernstein, D. J. How to Stretch Random Functions: The Security of Protected Counter Sums. *J. Cryptology 12*, 3 (1999), 185–192.

[38] Bernstein, D. J. Stronger Security Bounds for Wegman-Carter-Shoup Authenticators. In Cramer [63], pp. 164–180.

[39] Bernstein, D. J. The Poly1305-AES Message-Authentication Code. In *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers* (2005), H. Gilbert and H. Handschuh, Eds., vol. 3557 of *Lecture Notes in Computer Science*, Springer, pp. 32–49.

[40] Bernstein, D. J., and Lange, T. Non-uniform Cracks in the Concrete: The Power of Free Precomputation. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II* (2013), K. Sako and P. Sarkar, Eds., vol. 8270 of *Lecture Notes in Computer Science*, Springer, pp. 321–340.

[41] Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Pironti, A., and Strub, P. Triple Handshakes and Cookie Cutters: Breaking and Fixing Authentication over TLS. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014* (2014), IEEE Computer Society, pp. 98–113. `http://www.mitls.org/downloads/tlsauth.pdf`.

[42] Bhargavan, K., Leurent, G., Cadé, D., Blanchet, B., Paraskevopoulou, Z., Hriţcu, C., Dénès, M., Lampropoulos, L., Pierce, B. C., Delignat-Lavaud, A., et al. Transcript Collision Attacks: Breaking Authentication in TLS, IKE, and SSH. In *Network and Distributed System Security Symposium–NDSS 2016* (2016). `http://www.mitls.org/downloads/transcript-collisions.pdf`.

[43] Bhaumik, R., and Nandi, M. An Inverse-Free Single-Keyed Tweakable Enciphering Scheme. In Iwata and Cheon [97], pp. 159–180.

[44] Bhaumik, R., and Nandi, M. Revisiting Turning Online Cipher Off. Cryptology ePrint Archive, Report 2015/813, 2015.

[45] Biryukov, A., Ed. *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers* (2007), vol. 4593 of *Lecture Notes in Computer Science*, Springer.

[46] Black, J., Cochran, M., and Highland, T. A Study of the MD5 Attacks: Insights and Improvements. In *FSE* (2006), M. J. B. Robshaw, Ed., vol. 4047 of *Lecture Notes in Computer Science*, Springer, pp. 262–277.

[47] Black, J., and Rogaway, P. A Block-Cipher Mode of Operation for Parallelizable Message Authentication. In Knudsen [109], pp. 384–397.

[48] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C. PRESENT: An Ultra-Lightweight Block Cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings* (2007), P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *Lecture Notes in Computer Science*, Springer, pp. 450–466.

[49] Bogdanov, A., Mendel, F., Regazzoni, F., Rijmen, V., and Tischhauser, E. ALE: AES-Based Lightweight Authenticated Encryption. In *FSE 2013* (2013), S. Moriai, Ed., vol. 8424 of *Lecture Notes in Computer Science*, Springer, pp. 447–466.

[50] Boldyreva, A., Degabriele, J. P., Paterson, K. G., and Stam, M. Security of Symmetric Encryption in the Presence of Ciphertext Fragmentation. In *EUROCRYPT 2012* (2012), D. Pointcheval and T. Johansson, Eds., vol. 7237 of *Lecture Notes in Computer Science*, Springer, pp. 682–699.

[51] Boldyreva, A., Degabriele, J. P., Paterson, K. G., and Stam, M. On Symmetric Encryption with Distinguishable Decryption Failures. In *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers* (2013), S. Moriai, Ed., vol. 8424 of *Lecture Notes in Computer Science*, Springer, pp. 367–390.

[52] Boldyreva, A., Degabriele, J. P., Paterson, K. G., and Stam, M. On Symmetric Encryption with Distinguishable Decryption Failures. Cryptology ePrint Archive, Report 2013/433, 2013.

[53] Boldyreva, A., Degabriele, J. P., Paterson, K. G., and Stam, M. Security of Symmetric Encryption in the Presence of Ciphertext Fragmentation. Cryptology ePrint Archive, Report 2015/059, 2015.

[54] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E. B., Knezevic, M., Knudsen, L. R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S. S., and Yalçın, T. PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In Wang and Sako [172], pp. 208–225.

[55] Borisov, N., Goldberg, I., and Wagner, D. Intercepting mobile communications: the insecurity of 802.11. In *MOBICOM* (2001), C. Rose, Ed., ACM, pp. 180–189.

[56] Cannière, C. D., Dunkelman, O., and Knezevic, M. KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings* (2009), C. Clavier and K. Gaj, Eds., vol. 5747 of *Lecture Notes in Computer Science*, Springer, pp. 272–288.

[57] Cantero, H. M., Peter, S., Bushing, and Segher. Console Hacking 2010 – PS3 Epic Fail. 27th Chaos Communication Congress, December 2010.

[58] Canvel, B., Hiltgen, A. P., Vaudenay, S., and Vuagnoux, M. Password Interception in a SSL/TLS Channel. In *CRYPTO* (2003), D. Boneh, Ed., vol. 2729 of *Lecture Notes in Computer Science*, Springer, pp. 583–599.

[59] Chakraborty, D., and Sarkar, P. HCH: A New Tweakable Enciphering Scheme Using the Hash-Counter-Hash Approach. *IEEE Transactions on Information Theory 54*, 4 (2008), 1683–1699.

[60] Chang, D., and Nandi, M. A Short Proof of the PRP/PRF Switching Lemma. Cryptology ePrint Archive, Report 2008/078, 2008.

[61] Cid, C., and Rechberger, C., Eds. *Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers* (2015), vol. 8540 of *Lecture Notes in Computer Science*, Springer.

[62] Coppersmith, D., Ed. *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings* (1995), vol. 963 of *Lecture Notes in Computer Science*, Springer.

[63] Cramer, R., Ed. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings* (2005), vol. 3494 of *Lecture Notes in Computer Science*, Springer.

[64] Daemen, J. *Hash Function and Cipher Design: Strategies Based on Linear and Differential Cryptanalysis.* PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1995.

[65] Daemen, J., Peeters, M., Van Assche, G., and Rijmen, V. Nessie Proposal: Noekeon. First Open Nessie Workshop, 2000.

[66] Daemen, J., and Rijmen, V. AES proposal: Rijndael. First Advanced Encryption Standard (AES) Conference, 1998.

[67] Daemen, J., and Rijmen, V. *The Design of Rijndael: AES - The Advanced Encryption Standard.* Springer, 2002.

[68] Datta, N., and Yasuda, K. Generalizing PMAC Under Weaker Assumptions. In *Information Security and Privacy - 20th Australasian Conference, ACISP 2015, Brisbane, QLD, Australia, June 29 - July 1, 2015, Proceedings* (2015), E. Foo and D. Stebila, Eds., vol. 9144 of *Lecture Notes in Computer Science*, Springer, pp. 433–450.

[69] Desai, A. New Paradigms for Constructing Symmetric Encryption Schemes Secure against Chosen-Ciphertext Attack. In *CRYPTO* (2000), M. Bellare, Ed., vol. 1880 of *Lecture Notes in Computer Science*, Springer, pp. 394–412.

[70] Dodis, Y., and Pietrzak, K. Improving the Security of MACs Via Randomized Message Preprocessing. In Biryukov [45], pp. 414–433.

[71] Dodis, Y., Pietrzak, K., and Puniya, P. A New Mode of Operation for Block Ciphers and Length-Preserving MACs. In *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings* (2008), N. P. Smart, Ed., vol. 4965 of *Lecture Notes in Computer Science*, Springer, pp. 198–219.

[72] Dodis, Y., and Steinberger, J. P. Message Authentication Codes from Unpredictable Block Ciphers. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings* (2009), S. Halevi, Ed., vol. 5677 of *Lecture Notes in Computer Science*, Springer, pp. 267–285.

[73] Dodis, Y., and Steinberger, J. P. Domain Extension for MACs Beyond the Birthday Barrier. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings* (2011), K. G. Paterson, Ed., vol. 6632 of *Lecture Notes in Computer Science*, Springer, pp. 323–342.

[74] Dziembowski, S., and Pietrzak, K. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA* (2008), IEEE Computer Society, pp. 293–302.

[75] Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., and Walker, J. The Skein Hash Function Family, 2009. Submission to NIST's SHA-3 competition.

[76] Fleischmann, E., Forler, C., and Lucks, S. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In *FSE* (2012), A. Canteaut, Ed., vol. 7549 of *Lecture Notes in Computer Science*, Springer, pp. 196–215.

[77] Fleischmann, E., Forler, C., Lucks, S., and Wenzel, J. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. Cryptology ePrint Archive, Report 2011/644, 2011.

[78] Fouque, P.-A., Joux, A., Martinet, G., and Valette, F. Authenticated On-Line Encryption. In *Selected Areas in Cryptography* (2003), M. Matsui and R. J. Zuccherato, Eds., vol. 3006 of *Lecture Notes in Computer Science*, Springer, pp. 145–159.

[79] Garay, J. A., and Gennaro, R., Eds. *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I* (2014), vol. 8616 of *Lecture Notes in Computer Science*, Springer.

[80] Gaži, P., Pietrzak, K., and Rybár, M. The Exact PRF-Security of NMAC and HMAC. In Garay and Gennaro [79], pp. 113–130.

[81] Gennaro, R., and Robshaw, M., Eds. *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA,*

USA, August 16-20, 2015, Proceedings, Part I (2015), vol. 9215 of *Lecture Notes in Computer Science*, Springer.

[82] Gérard, B., Grosso, V., Naya-Plasencia, M., and Standaert, F. Block Ciphers That Are Easier to Mask: How Far Can We Go? In *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings* (2013), G. Bertoni and J. Coron, Eds., vol. 8086 of *Lecture Notes in Computer Science*, Springer, pp. 383–399.

[83] Goldwasser, S., and Micali, S. Probabilistic Encryption. *J. Comput. Syst. Sci. 28*, 2 (1984), 270–299.

[84] Gong, Z., Nikova, S., and Law, Y. W. KLEIN: A New Family of Lightweight Block Ciphers. In *RFID. Security and Privacy - 7th International Workshop, RFIDSec 2011, Amherst, USA, June 26-28, 2011, Revised Selected Papers* (2011), A. Juels and C. Paar, Eds., vol. 7055 of *Lecture Notes in Computer Science*, Springer, pp. 1–18.

[85] Gostev, A. The Flame: Questions and Answers. `https://www.securelist.com/en/blog/208193522/The_Flame_Questions_and_Answers`, May 2012. Date accessed 2016.03.04.

[86] Grosso, V., Leurent, G., Standaert, F., and Varici, K. LS-Designs: Bitslice Encryption for Efficient Masked Software Implementations. In Cid and Rechberger [61], pp. 18–37.

[87] Gueron, S. AES-GCM software performance on the current high end CPUs as a performance baseline for CAESAR competition. Directions in Authenticated Ciphers (DIAC), 2013.

[88] Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M. J. B. The LED Block Cipher. In Preneel and Takagi [145], pp. 326–341.

[89] Hall, C., Wagner, D., Kelsey, J., and Schneier, B. Building PRFs from PRPs. In *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings* (1998), H. Krawczyk, Ed., vol. 1462 of *Lecture Notes in Computer Science*, Springer, pp. 370–389.

[90] Handschuh, H., and Preneel, B. Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings* (2008), D. Wagner, Ed., vol. 5157 of *Lecture Notes in Computer Science*, Springer, pp. 144–161.

[91] Hoang, V. T., Katz, J., and Malozemoff, A. J. Automated Analysis and Synthesis of Authenticated Encryption Schemes. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015* (2015), I. Ray, N. Li, and C. Kruegel, Eds., ACM, pp. 84–95.

[92] Hoang, V. T., Krovetz, T., and Rogaway, P. Robust Authenticated-Encryption: AEZ and the Problem that it Solves. *IACR Cryptology ePrint Archive 2014* (2014), 793.

[93] Hoang, V. T., Reyhanitabar, R., Rogaway, P., and Vizár, D. Online Authenticated-Encryption and its Nonce-Reuse Misuse-Resistance. In Gennaro and Robshaw [81], pp. 493–517.

[94] Hong, D., Lee, J., Kim, D., Kwon, D., Ryu, K. H., and Lee, D. LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors. In *Information Security Applications - 14th International Workshop, WISA 2013, Jeju Island, Korea, August 19-21, 2013, Revised Selected Papers* (2013), Y. Kim, H. Lee, and A. Perrig, Eds., vol. 8267 of *Lecture Notes in Computer Science*, Springer, pp. 3–27.

[95] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., and Chee, S. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings* (2006), L. Goubin and M. Matsui, Eds., vol. 4249 of *Lecture Notes in Computer Science*, Springer, pp. 46–59.

[96] Impagliazzo, R., and Rudich, S. Limits on the Provable Consequences of One-Way Permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA* (1989), D. S. Johnson, Ed., ACM, pp. 44–61.

[97] Iwata, T., and Cheon, J. H., Eds. *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II* (2015), vol. 9453 of *Lecture Notes in Computer Science*, Springer.

[98] Iwata, T., and Kurosawa, K. Stronger Security Bounds for OMAC, TMAC, and XCBC. In *Progress in Cryptology - INDOCRYPT 2003, 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003, Proceedings* (2003), T. Johansson and S. Maitra, Eds., vol. 2904 of *Lecture Notes in Computer Science*, Springer, pp. 402–415.

[99] Iwata, T., and Yasuda, K. BTM: A Single-Key, Inverse-Cipher-Free Mode for Deterministic Authenticated Encryption. In *Selected Areas in Cryptography* (2009), M. J. Jacobson Jr, V. Rijmen, and R. Safavi-Naini, Eds., vol. 5867 of *Lecture Notes in Computer Science*, Springer, pp. 313–330.

[100] Iwata, T., and Yasuda, K. HBS: A Single-Key Mode of Operation for Deterministic Authenticated Encryption. In *FSE* (2009), O. Dunkelman, Ed., vol. 5665 of *Lecture Notes in Computer Science*, Springer, pp. 394–415.

[101] Jean, J., Nikolic, I., and Peyrin, T. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II* (2014), P. Sarkar and T. Iwata, Eds., vol. 8874 of *Lecture Notes in Computer Science*, Springer, pp. 274–288.

[102] Journault, A., Standaert, F.-X., and Varici, K. Improving the Security and Efficiency of Block Ciphers based on LS-Designs. proceedings of the 9th International Workshop on Coding and Cryptography, WCC 2015, 2015.

[103] Joux, A. Authentication Failures in NIST Version of GCM. `http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/Joux_comments.pdf`, 2006. Date accessed 2016.02.20.

[104] Joux, A., Martinet, G., and Valette, F. Blockwise-Adaptive Attackers: Revisiting the (In)Security of Some Provably Secure Encryption Models: CBC, GEM, IACBC. In Yung [181], pp. 17–30.

[105] Jovanovic, P., Luykx, A., and Mennink, B. Beyond 2 c/2 security in sponge-based authenticated encryption modes. In Sarkar and Iwata [160], pp. 85–104.

[106] Karakoç, F., Demirci, H., and Harmanci, A. E. ITUbee: A Software Oriented Lightweight Block Cipher. In *Lightweight Cryptography for Security and Privacy - Second International Workshop, LightSec 2013, Gebze, Turkey, May 6-7, 2013, Revised Selected Papers* (2013), G. Avoine and O. Kara, Eds., vol. 8162 of *Lecture Notes in Computer Science*, Springer, pp. 16–27.

[107] Katz, J., and Yung, M. Complete characterization of security notions for probabilistic private-key encryption. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23,*

2000, Portland, OR, USA (2000), F. F. Yao and E. M. Luks, Eds., ACM, pp. 245–254.

[108] Katz, J., and Yung, M. Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation. In *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings* (2000), B. Schneier, Ed., vol. 1978 of *Lecture Notes in Computer Science*, Springer, pp. 284–299.

[109] Knudsen, L. R., Ed. *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings* (2002), vol. 2332 of *Lecture Notes in Computer Science*, Springer.

[110] Kohno, T. Attacking and repairing the winZip encryption scheme. In *ACM Conference on Computer and Communications Security* (2004), V. Atluri, B. Pfitzmann, and P. D. McDaniel, Eds., ACM, pp. 72–81.

[111] Krovetz, T., and Rogaway, P. The Software Performance of Authenticated-Encryption Modes. In *FSE* (2011), A. Joux, Ed., vol. 6733 of *Lecture Notes in Computer Science*, Springer, pp. 306–327.

[112] Laywine, C. F., and Mullen, G. L. *Discrete mathematics using Latin squares*, vol. 49. John Wiley & Sons, 1998.

[113] Leander, G., Paar, C., Poschmann, A., and Schramm, K. New Lightweight DES Variants. In Biryukov [45], pp. 196–210.

[114] Lenstra, A. K., Hughes, J. P., Augier, M., Bos, J. W., Kleinjung, T., and Wachter, C. Public Keys. In *CRYPTO* (2012), R. Safavi-Naini and R. Canetti, Eds., vol. 7417 of *Lecture Notes in Computer Science*, Springer, pp. 626–642.

[115] Lim, C. H., and Korkishko, T. mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In *Information Security Applications, 6th International Workshop, WISA 2005, Jeju Island, Korea, August 22-24, 2005, Revised Selected Papers* (2005), J. Song, T. Kwon, and M. Yung, Eds., vol. 3786 of *Lecture Notes in Computer Science*, Springer, pp. 243–258.

[116] Liskov, M., Rivest, R. L., and Wagner, D. Tweakable Block Ciphers. In Yung [181], pp. 31–46.

[117] Luby, M., and Rackoff, C. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM J. Comput. 17*, 2 (1988), 373–386.

[118] Lucks, S. The Sum of PRPs Is a Secure PRF. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding* (2000), B. Preneel, Ed., vol. 1807 of *Lecture Notes in Computer Science*, Springer, pp. 470–484.

[119] Luykx, A., Preneel, B., Szepieniec, A., and Yasuda, K. On the Influence of Message Length in PMAC's Security Bounds. In *Advances in Cryptology - EUROCRYPT 2016* (Vienna,AT, 2016), J.-S. Coron and M. Fischlin, Eds., Lecture Notes in Computer Science, Springer-Verlag, p. 30.

[120] Luykx, A., Preneel, B., Szepieniec, A., and Yasuda, K. On the Influence of Message Length in PMAC's Security Bounds. Cryptology ePrint Archive, Report 2016/185, 2016.

[121] Luykx, A., Preneel, B., Tischhauser, E., and Yasuda, K. A MAC Mode for Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2016/190, 2016.

[122] Luykx, A., Preneel, B., Tischhauser, E., and Yasuda, K. A MAC Mode for Lightweight Block Ciphers. In *Fast Software Encryption, FSE 2016* (Bochum,DE, 2016), Lecture Notes in Computer Science, Springer-Verlag, p. 20.

[123] Maurer, U. M. Indistinguishability of random systems. In Knudsen [109], pp. 110–132.

[124] Maurer, U. M., and Sjödin, J. Single-Key AIL-MACs from Any FIL-MAC. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings* (2005), L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds., vol. 3580 of *Lecture Notes in Computer Science*, Springer, pp. 472–484.

[125] McGrew, D. A., and Viega, J. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In *INDOCRYPT* (2004), A. Canteaut and K. Viswanathan, Eds., vol. 3348 of *Lecture Notes in Computer Science*, Springer, pp. 343–355.

[126] Minematsu, K. How to Thwart Birthday Attacks against MACs via Small Randomness. In *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers* (2010), S. Hong and T. Iwata, Eds., vol. 6147 of *Lecture Notes in Computer Science*, Springer, pp. 230–249.

[127] Minematsu, K. Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions. In Nguyen and Oswald [141], pp. 275–292.

[128] Minematsu, K., and Matsushima, T. New Bounds for PMAC, TMAC, and XCBC. In Biryukov [45], pp. 434–451.

[129] Mouha, N., and Luykx, A. Multi-key security: The even-mansour construction revisited. In Gennaro and Robshaw [81], pp. 209–223.

[130] Namprempre, C., Rogaway, P., and Shrimpton, T. Reconsidering Generic Composition. In Nguyen and Oswald [141], pp. 257–274.

[131] Nandi, M. Improved security analysis for OMAC as a pseudorandom function. *J. Mathematical Cryptology 3*, 2 (2009), 133–148.

[132] Nandi, M. Forging Attack on COBRA. Cryptographic Competitions Google Group, 2014.

[133] Nandi, M. Forging Attacks on Two Authenticated Encryption Schemes COBRA and POET. In Sarkar and Iwata [160], pp. 126–140.

[134] Nandi, M. Forging Attacks on two Authenticated Encryptions COBRA and POET. Cryptology ePrint Archive, Report 2014/363, 2014.

[135] Nandi, M. XLS is Not a Strong Pseudorandom Permutation. In Sarkar and Iwata [160], pp. 478–490.

[136] Nandi, M. On the Optimality of Non-Linear Computations of Length-Preserving Encryption Schemes. In Iwata and Cheon [97], pp. 113–133.

[137] Nandi, M. Revisiting Security Claims of XLS and COPA. *IACR Cryptology ePrint Archive 2015* (2015), 444.

[138] Nandi, M., and Mandal, A. Improved security analysis of PMAC. *J. Mathematical Cryptology 2*, 2 (2008), 149–162.

[139] National Institute of Standards and Technology. DES Modes of Operation. FIPS 81, December 1980.

[140] Needham, R. M., and Wheeler, D. J. Tea extensions, 1997.

[141] Nguyen, P. Q., and Oswald, E., Eds. *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings* (2014), vol. 8441 of *Lecture Notes in Computer Science*, Springer.

[142] Paterson, K. G., and AlFardan, N. J. Plaintext-Recovery Attacks Against Datagram TLS. In *NDSS* (2012), The Internet Society.

[143] Petrank, E., and Rackoff, C. CBC MAC for Real-Time Data Sources. *JOURNAL OF CRYPTOLOGY 13* (1997), 315–338.

[144] Pietrzak, K. A Tight Bound for EMAC. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II* (2006), M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of *Lecture Notes in Computer Science*, Springer, pp. 168–179.

[145] Preneel, B., and Takagi, T., Eds. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings* (2011), vol. 6917 of *Lecture Notes in Computer Science*, Springer.

[146] Preneel, B., and van Oorschot, P. C. MDx-MAC and Building Fast MACs from Hash Functions. In Coppersmith [62], pp. 1–14.

[147] Rabin, M. O. Transaction protection by beacons. *Journal of Computer and System Sciences 27*, 2 (1983), 256 – 267.

[148] Ray, M., and Dispensa, S. Renegotiating TLS. `https://kryptera.se/Renegotiating%20TLS.pdf`. Date accessed 2016.03.03.

[149] Ristenpart, T., and Rogaway, P. How to Enrich the Message Space of a Cipher. In Biryukov [45], pp. 101–118.

[150] Ristenpart, T., and Rogaway, P. How to Enrich the Message Space of a Cipher. In *FSE 2007* (2007), A. Biryukov, Ed., vol. 4593 of *Lecture Notes in Computer Science*, Springer, pp. 101–118.

[151] Rivest, R. L. The RC5 Encryption Algorithm. In *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings* (1994), B. Preneel, Ed., vol. 1008 of *Lecture Notes in Computer Science*, Springer, pp. 86–96.

[152] Rogaway, P. Method and apparatus for realizing a parallelizable variable-input-length pseudorandom function, Sept. 5 2001. US Patent App. 09/948,084.

[153] Rogaway, P. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In *ASIACRYPT* (2004), P. J. Lee, Ed., vol. 3329 of *Lecture Notes in Computer Science*, Springer, pp. 16–31.

[154] Rogaway, P. Nonce-Based Symmetric Encryption. In *FSE 2004* (2004), B. K. Roy and W. Meier, Eds., vol. 3017 of *Lecture Notes in Computer Science*, Springer, pp. 348–359.

[155] Rogaway, P., Bellare, M., Black, J., and Krovetz, T. OCB: a block-cipher mode of operation for efficient authenticated encryption. In *ACM Conference on Computer and Communications Security* (2001), M. K. Reiter and P. Samarati, Eds., ACM, pp. 196–205.

[156] Rogaway, P., and Shrimpton, T. A Provable-Security Treatment of the Key-Wrap Problem. In *EUROCRYPT 2006* (2006), S. Vaudenay, Ed., vol. 4004 of *Lecture Notes in Computer Science*, Springer, pp. 373–390.

[157] Rogaway, P., Wooding, M., and Zhang, H. The Security of Ciphertext Stealing. In *FSE 2012* (2012), A. Canteaut, Ed., vol. 7549 of *Lecture Notes in Computer Science*, Springer, pp. 180–195.

[158] Rogaway, P., and Zhang, H. Online Ciphers from Tweakable Blockciphers. In *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings* (2011), A. Kiayias, Ed., vol. 6558 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 237–249.

[159] Sako, K., and Sarkar, P., Eds. *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I* (2013), vol. 8269 of *Lecture Notes in Computer Science*, Springer.

[160] Sarkar, P., and Iwata, T., Eds. *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I* (2014), vol. 8873 of *Lecture Notes in Computer Science*, Springer.

[161] Schroeppel, R. The Hasty Pudding Cipher, 1998. Submission to NIST's AES competition.

[162] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., and Shirai, T. Piccolo: An Ultra-Lightweight Blockcipher. In Preneel and Takagi [145], pp. 342–357.

[163] Shirai, T., Shibutani, K., Akishita, T., Moriai, S., and Iwata, T. The 128-Bit Blockcipher CLEFIA (Extended Abstract). In Biryukov [45], pp. 181–195.

[164] Shoup, V. New Algorithms for Finding Irreducible Polynomials over Finite Fields. In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988* (1988), IEEE Computer Society, pp. 283–290.

[165] Shrimpton, T., and Terashima, R. S. A Modular Framework for Building Variable-Input-Length Tweakable Ciphers. In Sako and Sarkar [159], pp. 405–423.

[166] Standaert, F., Piret, G., Gershenfeld, N., and Quisquater, J. SEA: A Scalable Encryption Algorithm for Small Embedded Applications. In *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, April 19-21, 2006, Proceedings* (2006), J. Domingo-Ferrer, J. Posegga, and D. Schreckling, Eds., vol. 3928 of *Lecture Notes in Computer Science*, Springer, pp. 222–236.

[167] Suzaki, T., Minematsu, K., Morioka, S., and Kobayashi, E. TWINE : A Lightweight Block Cipher for Multiple Platforms. In *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers* (2012), L. R. Knudsen and H. Wu, Eds., vol. 7707 of *Lecture Notes in Computer Science*, Springer, pp. 339–354.

[168] Tsang, P. P., and Smith, S. W. Secure Cryptographic Precomputation with Insecure Memory. In *ISPEC 2008* (2008), L. Chen, Y. Mu, and W. Susilo, Eds., vol. 4991 of *Lecture Notes in Computer Science*, Springer, pp. 146–160.

[169] Tsang, P. P., Solomakhin, R. V., and Smith, S. W. Authenticated Streamwise On-line Encryption. Dartmouth Computer Science Technical Report TR2009-640, 2009.

[170] Vaudenay, S. Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS ... In Knudsen [109], pp. 534–546.

[171] Vernam, G. Secret signaling system, July 22 1919. US Patent 1,310,719.

[172] Wang, X., and Sako, K., Eds. *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings* (2012), vol. 7658 of *Lecture Notes in Computer Science*, Springer.

[173] Wang, X., and Yu, H. How to Break MD5 and Other Hash Functions. In Cramer [63], pp. 19–35.

[174] Wegman, M. N., and Carter, L. New Hash Functions and Their Use in Authentication and Set Equality. *J. Comput. Syst. Sci. 22*, 3 (1981), 265–279.

[175] Wu, H. The Misuse of RC4 in Microsoft Word and Excel. Cryptology ePrint Archive, Report 2005/007, 2005.

[176] Wu, W., and Zhang, L. LBlock: A Lightweight Block Cipher. In *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings* (2011), J. Lopez and G. Tsudik, Eds., vol. 6715 of *Lecture Notes in Computer Science*, pp. 327–344.

[177] Yang, G., Zhu, B., Suder, V., Aagaard, M. D., and Gong, G. The Simeck Family of Lightweight Block Ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings* (2015), T. Güneysu and H. Handschuh, Eds., vol. 9293 of *Lecture Notes in Computer Science*, Springer, pp. 307–329.

[178] Yasuda, K. The Sum of CBC MACs Is a Secure PRF. In *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings* (2010), J. Pieprzyk, Ed., vol. 5985 of *Lecture Notes in Computer Science*, Springer, pp. 366–381.

[179] Yasuda, K. A New Variant of PMAC: Beyond the Birthday Bound. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings* (2011), P. Rogaway, Ed., vol. 6841 of *Lecture Notes in Computer Science*, Springer, pp. 596–609.

[180] Yasuda, K. PMAC with Parity: Minimizing the Query-Length Influence. In *Topics in Cryptology - CT-RSA 2012 - The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 - March 2, 2012. Proceedings* (2012), O. Dunkelman, Ed., vol. 7178 of *Lecture Notes in Computer Science*, Springer, pp. 203–214.

[181] Yung, M., Ed. *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings* (2002), vol. 2442 of *Lecture Notes in Computer Science*, Springer.

[182] Zhang, H. Length-Doubling Ciphers and Tweakable Ciphers. In *Applied Cryptography and Network Security - 10th International Conference,*

ACNS 2012, Singapore, June 26-29, 2012. Proceedings (2012), F. Bao, P. Samarati, and J. Zhou, Eds., vol. 7341 of *Lecture Notes in Computer Science*, Springer, pp. 100–116.

[183] Zhang, L., Wu, W., Sui, H., and Wang, P. 3kf9: Enhancing 3GPP-MAC beyond the Birthday Bound. In Wang and Sako [172], pp. 296–312.

[184] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., and Verbauwhede, I. RECTANGLE: A Bit-slice Lightweight Block Cipher Suitable for Multiple Platforms. Cryptology ePrint Archive, Report 2014/084, 2014.

[185] Zhang, Y. Using an Error-Correction Code for Fast, Beyond-Birthday-Bound Authentication. In *Topics in Cryptology — CT-RSA 2015*, K. Nyberg, Ed., vol. 9048 of *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 291–307.

CV

Education

KU Leuven, Faculty of Engineering Science, Louvain, Belgium 2012 – 2016
PhD in Cryptography
Adviser: Bart Preneel
Funded by a Fellowship from IWT-Vlaanderen

Research Internship at NTT Secure Platform Laboratories, Japan
January – July 2015

Research Visit at University of Haifa, Israel
October – November 2015

Research Visit at DTU, Denmark
April 2016

Belgian expert delegate to the ISO/IEC JTC1/SC27/WG2
September 2015 – Present

KU Leuven, Faculty of Engineering Science, Louvain, Belgium 2010 – 2012
Master's in Mathematical Engineering
Graduated magna cum laude
Thesis: *The Scope Of Indifferentiability and An Application To BLAKE*
Advisers: Bart Preneel and Vincent Rijmen

Cornell University, College of Arts and Sciences, Ithaca, NY 2006 – 2010
Bachelor's in Mathematics
Graduated cum laude

Teaching Experience

KU Leuven, Faculty of Engineering Science

TA for Linear Algebra	Fall 2014, 2015
Supervision of Master student Laura Winnen	Fall 2013 – Spring 2014
TA for Cryptography and Network Security	Spring 2013, 2014, 2016
TA for Informatie-overdracht en -verwerking	Fall 2012, 2013

Cornell University Mathematics Department

Tutor at the Mathematics Support Center	Fall 2009

Cornell University Computer Science Department

TA for CS 2110 and 2111	Fall 2008 – Summer 2009
Consultant for CS 100 and 211	Spring 2007 – Spring 2008

Reviews

ACM Symposium on Theory of Computing (STOC)	2016
Australasian Conference on Information Security and Privacy (ACISP)	2015
Applied Cryptography and Network Security (ACNS)	2014
Asiacrypt	2013, 2014, 2015
Cryptology and Network Security (CANS)	2013
Crypto	2014, 2015, 2016
RSA Conference Cryptographers' Track (CT-RSA)	2014, 2015
Eurocrypt	2015, 2016
Fast Software Encryption (FSE)	2013, 2014, 2016
Indocrypt	2014
Information Security Conference (ISC)	2014
International Workshop on Security (IWSEC)	2013
Selected Areas in Cryptography (SAC)	2015
Usenix	2015

Talks

1. *On the Influence of Message Length in PMAC's Security Bounds*
   Eurocrypt 2016
   `http://ist.ac.at/eurocrypt2016/program.html`
   Vienna, Austria, May 11th, 2016
2. *A MAC Mode for Lightweight Block Ciphers*
   Fast Software Encryption 2016
   `https://fse.rub.de/program.html`
   Bochum, Germany, March 21st, 2016
3. *A MAC Mode for Lightweight Block Ciphers*
   COSIC Seminar
   Leuven, Belgium, March 17th, 2016
4. *Authenticated Encryption*
   School on Design for a Secure Internet of Things
   `https://www.cosic.esat.kuleuven.be/school-iot/index.shtml`
   Tenerife, Spain, January 27th, 2016
5. *The Limited Power of Verification Queries in Message Authentication and Authenticated Encryption*
   DIAC 2015: Directions in Authenticated Ciphers
   `http://www1.spms.ntu.edu.sg/~diac2015/`
   Singapore, September 29th, 2015
6. *Beyond $2^{c/2}$ Security in Sponge-Based Authenticated Encryption Modes*
   Asiacrypt 2014
   `http://des.cse.nsysu.edu.tw/asiacrypt2014/`
   Kaohsiung, Taiwan, December 8th, 2014
7. *How to Securely Release Unverified Plaintext in Authenticated Encryption*
   Asiacrypt 2014
   `http://des.cse.nsysu.edu.tw/asiacrypt2014/`
   Kaohsiung, Taiwan, December 8th, 2014

8. *How to Securely Release Unverified Plaintext in Authenticated Encryption*
   DIAC 2014: Directions in Authenticated Ciphers
   `http://2014.diac.cr.yp.to/`
   Santa Barbara, CA, USA, August 22nd, 2014
9. *Beyond $2^{c/2}$ Security in Sponge-Based Authenticated Encryption Modes*
   Design and security of cryptographic algorithms and devices for real-world applications
   `http://summerschool-croatia14.cs.ru.nl/index.shtml`
   Šibenik, Croatia, June 3rd, 2014
10. *COBRA: A Parallelizable Authenticated Online Cipher Without Block Cipher Inverse*
    Fast Software Encryption 2014
    `http://fse2014.isg.rhul.ac.uk/`
    London, UK, March 3rd, 2014
11. *Parallelizable and Authenticated Online Ciphers*
    Asiacrypt 2013
    `http://www.iacr.org/conferences/asiacrypt2013`
    Bangalore, India, December 3rd, 2013
12. *Parallelizable and Authenticated Online Ciphers*
    COSIC seminar
    Leuven, Belgium, November 29th, 2013
13. *APE(X): authenticated permutation-based encryption with extended security features*
    DIAC 2013: Directions in Authenticated Ciphers
    `http://2013.diac.cr.yp.to/`
    Chicago, USA, August 12th, 2013
14. *APE(X): Authenticated Permutation-Based Encryption with Extended Misuse Resistance*
    COSIC seminar
    Leuven, Belgium, August 9th, 2013

15. *Nonce-free Authenticated Encryption with Permutations*
    Ice Break Summer School
    `http://ice.mat.dtu.dk/`
    Reykjavik, Iceland, June 6th, 2013

Publications

1. Luykx A., Preneel B., Szepieniec A., Yasuda K. On the Influence of Message Length in PMAC's Security Bounds. In *Advances in Cryptology - EUROCRYPT 2016*, Lecture Notes in Computer Science, Springer-Verlag. To appear.

2. Luykx A., Preneel B., Tischhauser E., Yasuda K. A MAC Mode for Lightweight Block Ciphers. *Fast Software Encryption, FSE 2016*, Lecture Notes in Computer Science, Springer-Verlag. To appear.

3. Mouha N., Luykx A., Multi-Key Security: The Even-Mansour Construction Revisited. *Advances in Cryptology - CRYPTO 2015*, Lecture Notes in Computer Science, Springer-Verlag.

4. Luykx A., Mennink B., Preneel B., Winnen L. Two-Permutation-Based Hashing with Binary Mixing. *Journal of Mathematical Cryptology*, 2015.

5. Andreeva E., Bogdanov A., Luykx A., Mennink B., Mouha N., Yasuda K. How to Securely Release Unverified Plaintext in Authenticated Encryption. *Advances in Cryptology - ASIACRYPT 2014*, Lecture Notes in Computer Science, Springer-Verlag.

6. Jovanovic P., Luykx A., Mennink B. Beyond $2^{c/2}$ Security in Sponge-Based Authenticated Encryption Modes. *Advances in Cryptology - ASIACRYPT 2014*, Lecture Notes in Computer Science, Springer-Verlag.

7. Andreeva E., Bilgin B., Bogdanov A., Luykx A., Mennink B., Mouha N., Yasuda K. APE: Authenticated Permutation-Based Encryption for Lightweight Cryptography. *Fast Software Encryption, FSE 2014*, Lecture Notes in Computer Science, Springer-Verlag.

8. Andreeva E., Luykx A., Mennink B., Yasuda K. COBRA: A Parallelizable Authenticated Online Cipher Without Block Cipher Inverse. *Fast Software*

Encryption, FSE 2014, Lecture Notes in Computer Science, Springer-Verlag.

9. Andreeva E., Bogdanov A., Luykx A., Mennink B., Tischhauser E., Yasuda K. Parallelizable and Authenticated Online Ciphers. *Advances in Cryptology - ASIACRYPT 2013*, Lecture Notes in Computer Science, Springer-Verlag.

10. Andreeva E., Luykx A., Mennink B. Provable Security of BLAKE with Non-Ideal Compression Function. *Selected Areas in Cryptography, 19th Annual International Workshop, SAC 2012*, Lecture Notes in Computer Science, Springer-Verlag.