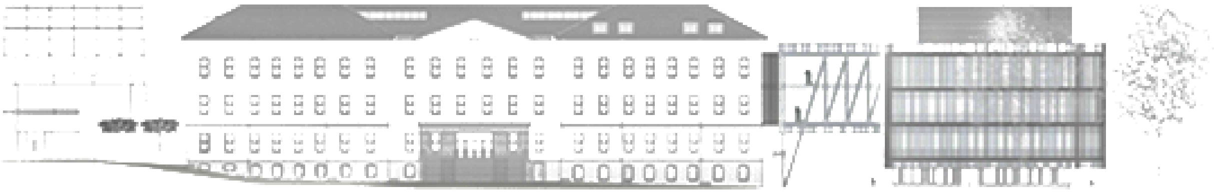


(The exact security of) Message Authentication Codes

by

Michal Rybár
June, 2017

A thesis presented to the
Graduate School
of the
Institute of Science and Technology Austria, Klosterneuburg, Austria
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

I|S|T AUSTRIA
Institute of Science and Technology

I hereby declare that this thesis is my own work and that it does not contain other people's work without this being so stated; this thesis does not contain my previous work without this being stated, and the bibliography contains all the literature that I used in writing the dissertation.
I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee, and that this thesis has not been submitted for a higher degree to any other university or institution.
I certify that any republication of materials presented in this thesis has been approved by the relevant publishers and co-authors.

Signature: ______________________

Michal Rybár
June, 2017

Abstract

In this thesis we discuss the exact security of message authentications codes HMAC, NMAC, and PMAC. NMAC is a mode of operation which turns a fixed input-length keyed hash function f into a variable input-length function. A practical single-key variant of NMAC called HMAC is a very popular and widely deployed message authentication code (MAC). PMAC is a block-cipher based mode of operation, which also happens to be the most famous fully parallel MAC.

NMAC was introduced by Bellare, Canetti and Krawczyk Crypto'96, who proved it to be a secure pseudorandom function (PRF), and thus also a MAC, under two assumptions. Unfortunately, for many instantiations of HMAC one of them has been found to be wrong. To restore the provable guarantees for NMAC, Bellare [Crypto'06] showed its security without this assumption.

PMAC was introduced by Black and Rogaway at Eurocrypt 2002. If instantiated with a pseudorandom permutation over n-bit strings, PMAC constitutes a provably secure variable input-length PRF. For adversaries making q queries, each of length at most ℓ (in n-bit blocks), and of total length $\sigma \leq q\ell$, the original paper proves an upper bound on the distinguishing advantage of $O(\sigma^2/2^n)$, while the currently best bound is $O(q\sigma/2^n)$. In this work we show that this bound is tight by giving an attack with advantage $\Omega(q^2\ell/2^n)$. In the PMAC construction one initially XORs a mask to every message block, where the mask for the ith block is computed as $\tau_i := \gamma_i \cdot L$, where L is a (secret) random value, and γ_i is the i-th codeword of the Gray code. Our attack applies more generally to any sequence of γ_i's which contains a large coset of a subgroup of $GF(2^n)$.

As for NMAC, our first contribution is a simpler and *uniform* proof: If f is an ε-secure PRF (against q queries) and a δ-*non-adaptively* secure PRF (against q queries), then $\mathsf{NMAC}^{\mathsf{f}}$ is an $(\varepsilon + \ell q\delta)$-secure PRF against q queries of length at most ℓ blocks each. We also show that this $\varepsilon + \ell q\delta$ bound is basically tight by constructing an f for which an attack with advantage $\ell q\delta$ exists.

Moreover, we analyze the PRF-security of a modification of NMAC called NI by An and Bellare that avoids the constant rekeying on multi-block messages in NMAC and allows for an information-theoretic analysis. We carry out such an analysis, obtaining a tight $\ell q^2/2^c$ bound for this step, improving over the trivial bound of $\ell^2q^2/2^c$.

Finally, we investigate, if the security of PMAC can be further improved by using τ_i's that are k-wise independent, for $k > 1$ (the original has $k = 1$). We observe that the security of PMAC will not increase in general if $k = 2$, and then prove that the security increases to $O(q^2/2^n)$, if the $k = 4$. Due to simple extension attacks, this is the best bound one can hope for, using any distribution on the masks. Whether $k = 3$ is already sufficient to get this level of security is left as an open problem.

Keywords: Message authentication codes, Pseudorandom functions, HMAC, PMAC.

About the Author

Michal Rybár decided to spend his university life abroad and therefore spent four years studying Mathematics and Computer Science at the University of Bristol, United Kingdom. He finished his efforts with First Class Honours and received the degree Master in Science, MSci. His specialization was in Pure Mathematics and Cryptography. After the completion of his master studies in July, Michal decided to further pursue his interest in science and therefore joined the Graduate School at the Institute of Science and Technology Austria, IST, in September 2012.

After spending one year in the multidisciplinary envirionmnent, he affiliated with the group of Krzysztof Pietrzak, a cryptographer. Michal's thesis proposal was on Provably Secure Authentication and he passed his qualifying exam in July 2013. During his PhD studies he focused mainly on Message Authentication Codes, publishing his joint work at Crypto 2014 and FSE 2017 conferences. The latter publication was chosen a runner-up for the best paper award. In addition, he published his joint work on differential privacy, and memory hard functions.

Last, but not least, Michal took a leave of 6 months during his studies to serve as a caretaker of a beautiful water mill in Kvačianska dolina, Slovakia, which is listed as national cultural heritage of Slovakia. Moreover, he served as a teaching assistant for the Computational Complexity course in 2014. He also took part in the comunity life of IST by helping to organize both the institute retreat, as well as the PhD retreat in 2013.

List of Publications

1. [*] Gaži, Peter, Krzysztof Pietrzak, and Michal Rybár. “The exact PRF-security of NMAC and HMAC.” In International Cryptology Conference, pp. 113-130. Springer Berlin Heidelberg, 2014. [24]

2. Yu, Fei, Michal Rybár, Caroline Uhler, and Stephen E. Fienberg. “Differentially-private logistic regression for detecting multiple-SNP association in GWAS databases.” In International Conference on Privacy in Statistical Databases, pp. 170-184. Springer International Publishing, 2014. [67]

3. Alwen, Joël, Peter Gaži, Chethan Kamath, Karen Klein, Georg Osang, Krzysztof Pietrzak, Leonid Reyzin, Michal Rolinek, and Michal Rybár. “On the Memory-Hardness of Data-Independent Password-Hashing Functions.” In Cryptology ePrint Archive, Report 2016/783, 2016. [3]

4. [†] Peter Gaži, Krzysztof Pietrzak, and Michal Rybár. “The Exact Security of PMAC.” IACR Transactions on Symmetric Cryptology 2016.2 (2017): 145-161. [26]

[*] Appears in the thesis.

[†] Appears in the thesis.

Table of Contents

List of Figures

List of Abbreviations

AES Advanced Encryption Standard

CBC Cipher Block Chaining

CBC-MAC Cipher Block Chaining Message Authentication Code

CDH Computational Diffie-Hellman

CIA Confidentiality, Integrity, Authenticity

DDH Decisional Diffie-Hellman

ECBC Encrypted Cipher Block Chaining

GHMAC Generalised Hash-based Message Authentication Code

GNMAC Generalised Nested Message Authentication Code

HMAC Hash-based Message Authentication Code

IKE Internet Key Exchange

IP Internet Protocol

LPN Learning Parity with Noise

MAC Message Authentication Code

MD Merkle-Damgard

MIM Man-in-the-middle

NA Non-adaptive

NI Nested Iterated transform

NMAC Nested Message Authentication Code

OCB Offset-codebook

OMAC One-key Message Authentication Code

PF Prefix-free

PMAC Parallelizable Message Authentication Code

PRF Pseudo-random Function

PRP Pseudo-random Permutation

RFID Radio-frequency identification

SHA Secure Hash Algorithm

SPMAC Simplified Parallelizable Message Authentication Code

SSH Secure Shell

SSL Secure Socket Layer

TLS Transport Layer Security

URF Uniformly Random Function

WCR Weakly Collision Resistant

1 Introduction

In this thesis, we explore the world of Cryptography. According to the Oxford Dictionary, Cryptography is “The art of writing or solving codes” [19]. Historically, it was mainly concerned with encryption, the art of making messages unreadable to everyone, but the intended receiver. This early stage of the field is very well documented in an extensive book by David Kahn, called The Codebreakers [34]. The modern history of cryptography started in the 1970s, most notably with the famous paper by Diffie and Hellman with an accurate title “New Directions in Cryptography” [20]. Nowadays, cryptographers define their field as “the study of mathematical techniques for securing digital information, systems, and distributed computations against adversarial attacks” [35]. Alternative definition could also say that “cryptography is concerned with the conceptualization, definition, and construction of computing systems that address security concerns” [27]. It is a rapidly growing field that attracts more and more researchers every year, while constantly finding new challenges and improving older constructions. In this thesis we focus on the latter - we go back and look at two cases of older constructions, and provide them with better security analysis. One of them, HMAC, is still widely deployed in many applications (especially in TLS, the most popular protocol for secure communication over the internet), and this fact provides the main source of motivation for this thesis.

Historically, cryptography was more of an art than science. Any (encryption) scheme usually had a very short lifespan. It was secure only until somebody discovered its “trick”- so called security by obscurity (a scheme is secure, because nobody really knows how it works). However, once this happened, it was not possible to use such a scheme anymore - only to change its design, or come up with a new one. Alternatively, these schemes could have been also broken by basic cryptanalysis (for example analyzing the secret text). Even more advanced schemes, like the Enigma machine, were *believed* to be secure, but without any formal guarantee. As you will see in Section 1.1, nowadays the situation is quite different.

1.1 Cryptography of today

In today’s world, cryptographic constructions are divided into two main classes - *symmetric* and *asymmetric*. In a symmetric construction, the two communicating parties, **A**lice and **B**ob, share the same key. In an asymmetric construction, each of them has a different key, but these two keys (called private and public) are related in some special way. Symmetric constructions are often simpler and computationally faster, but they have an obvious problem - key management. If Alice wants to communicate with a thousand parties, she needs to exchange and store exactly one thousand different keys in the symmetric setting. In the asymmetric setting, she needs to securely store just one, her private key, and look up public keys of others on a credentials server, for example. Therefore, asymmetric cryptography in theory solves the problem of key management, but at the expense of more expensive computation. However, the most effective solution is when these two classes work together to get the best out of both worlds. In this thesis, we work only in the symmetric setting.

Modern Cryptography goes around three main goals, CIA - confidentiality, integrity, and authenticity. Confidentiality, or privacy, is concerned with the secrecy of information. In other words, no adversary should be able to read information that we want to keep secret. The main tool of confidentiality is encryption, for example AES block cipher, or RSA public-key cryptosystem. The second goal, integrity, is about keeping data in their original form, unchanged. In this case, the adversary should not be able to modify protected data without detection. One of the main tools of integrity are Message Authentication Codes - MACs, discussed in Chapter 3, or cryptographic hash functions. The last goal in our list is authenticity, the task of knowing exactly who you are communicating with. In this case the adversary should not be able to assume the identity of a different entity. We can achieve authenticity through authentication protocols, for example.

The three goals can be achieved individually, but most often they work closely together, as in most cases we want to achieve them all at the same time. For example, let us consider an encryption scheme called *One-time pad*, which works by adding a fresh bit of the secret key to every bit of the message. It provably achieves perfect secrecy (confidentiality), but its integrity and authenticity can be easily breached, as any adversary can flip some bits of the secret message (and hence breach integrity), and also any adversary can send a random string as an encrypted message (this is considered a breach of authenticity). On the other hand, a MAC is a primitive that achieves both integrity and authenticity at the same time (but not confidentiality).

These following tools are used to achieve the three goals we have described:

Definitions Formal definitions are an essential cornerstone of modern cryptography. They are used to precisely define the goals we want to achieve and under exactly what assumptions. Cryptographic definitions need to be very clear, so everybody knows what they want to achieve, and more importantly, they need to know when they have actually achieved it. A bad or obscure definition can omit some important aspect of security and lead to faulty security proof, or allow for a simple attack that was completely missed by the definition.

Assumptions Assumptions are another essential building block of modern cryptography. They are well defined mathematical problems that are believed to be very difficult to solve (hard), yet this belief is without any formal proof. Any modern assumption must be clearly stated and accompanied by a rigorous definition. Often, with simpler and more "standard" assumptions come simpler schemes and they are adopted into practise more easily. As an example, we mention assumptions based on factoring - given a large number N, it should be computationally hard to find its prime number factorisation. The RSA cryptosystem is based on a variant of the factoring assumption, called the RSA problem.

Proofs Based on precise definitions, a cryptographic proof gives us security guarantees of a cryptographic scheme, with respect to some underlying assumptions. In simple words, a good proof says - this scheme is "secure", as long as the assumption it is build on holds. Security proofs are the basis of modern cryptography and any new cryptographic primitive must be accompanied by a rigorous proof in order to be taken seriously. Security proofs are the main step that transformed the field of cryptography from an art to a science discipline.

Kerckhoffs principle In 19th century, the linguist and cryptographer Auguste Kerckhoff stated roughly the following: “A good cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.” We state it here, because this statement brings together many of the basic principles described above. In order to be considered safe, any cryptographic scheme must be secure, even though its design is publicly known (unlike security by obscurity, where a scheme is intentionally kept secret). In other words, security of any scheme must rely solely on the security of an underlying cryptographic key. In fact, this is precisely what the community is aiming for - standardised cryptographic primitives that come out of a public competition after an extensive security analysis by the public. As an example, the AES block cipher and SHA-3 hash function are the products of such competitions.

Computational security Another very important tool in the cryptographer’s toolbox is the notion of computational security. Recognizing that perfect (or information-theoretic) security is an unnecessarily strong notion, we can build much more efficient schemes with the same level of *practical* security. We can say the following narrative: a perfectly secure scheme will leak no information to any adversary; a computationally secure scheme will leak only small amount of information to any adversary with bounded power. These bounds are expressed in terms of the number of computational operations and often can be translated into statements like ”using the currently fastest supercomputer on earth, it will take 147 years of nonstop computation to break this scheme”. In other words, for all practical purposes, a computationally secure scheme can be considered perfectly secure, and this is the approach almost all modern security proofs use.

Models Last, but not least, cryptographic models are a useful tool for writing proofs of security. They are in a way closely connected to assumptions. Most often, proofs are given using the *standard model*, without any idealised objects. However, it is often much simpler to assume that some *perfect* primitive exists, *e.g.* a random oracle or a random permutation (hence the random oracle, or the random permutation model, respectively). Even though these models were criticized for their applicability and real-world value, they passed the test of time and now are an important part of the cryptographic puzzle.

1.2 Future of Cryptography

The research in cryptography is more and more spread out. There are new primitives being defined, new applications proposed, and new schemes discovered. Through the many news reports on Bitcoin cryptocurrency and the cryptolocker virus, the public became much more aware of the field of cryptography and its goals. For symmetric cryptography, the current results consist mainly of improved security bounds and/or uniform proofs, where there have been none before. For asymmetric cryptography, there are still many applications waiting for new primitives (such as obfuscation). In fact, these new primitives are currently the most prolific part of the field.

The future also lies in more specialized primitives, whose application is narrower, but are much more efficient. Additionally, with new attacks, researchers will have to try to make their schemes more robust (such as leakage-resilient schemes try to counteract sophisticated cryptanalysis). As an example, we could mention self-driving cars that need ways of securing their systems against intrusion to protect them from malicious attacks.

Last, but definitely not least, with the possible rise of quantum computers, more research will have to be focused on quantum-secure cryptography. It is known for a long time that Shor's quantum factoring algorithm breaks the security of schemes based on factoring large numbers, most notably the RSA algorithm. However, these schemes are still widely deployed, because there are no quantum-secure alternatives with comparable efficiency.

1.3 Authentication

As we said before, authentication is one of the three most basic and profound tasks in the field of cryptography. According to Oxford Online Dictionary, "to authenticate" means to "have one's identity verified"[1]. A more cryptographic definition could say that to authenticate means to verify the identity of the party we are trying to communicate with. Imagine that Alice wishes to talk to Bob over some message channel. In an authentic communication, Alice is confident that she is talking to Bob, and Bob is confident that he is talking to Alice. Authenticity is breached, for example, if an adversary Eve manages to persuade Alice that she is talking to Bob, while impersonating him. Message forgery is another breach of authenticity, where Eve forges a new message which is accepted by either Bob, or Alice as genuine. Other failures of authentication could in some cases (depending on exact definition of authenticity) include message replay, where Eve sends some intercepted older message sent by Bob and this message is accepted by Alice as genuine.

Message Authentication Codes A Message Authentication Code, MAC, is a short piece of information, a tag, that is attached to the message to achieve two of the three basic cryptographic goals - integrity (the message has not been tampered with), and authenticity (we are guaranteed the origin of the message). We explore them further in Chapter 3.

Authentication Protocols A secret-key authentication protocol is a protocol between two parties, who share a secret key and want to verify their respective identities. Most often, it is one party wanting to authenticate its identity to a second party and in the context of the protocol, the parties are called the Prover and the Verifier. The security of an authentication protocol is traditionally defined against three types of adversaries - passive, active, and man-in-the-middle. The strongest one is the man-in-the-middle (MIM) attacker, while a passive attacker is the weakest.

In a secret-key authentication protocol the Prover $\mathcal{P}$ exchanges messages with the Verifier $\mathcal{V}$, trying to make him *accept*. In other words, $\mathcal{P}$ is trying to authenticate itself

[1] Oxford Dictionaries. Oxford University Press. http://oxforddictionaries.com/definition/english/authenticate (accessed December 05, 2016)

to $\mathcal{V}$. As mentioned earlier, there are three ways to define security of authentication protocols. The weakest notion, *passive* attack, proceeds in two phases. In the first phase, the adversary can observe any number of interactions between $\mathcal{P}$ and $\mathcal{V}$. In the second stage, the adversary tries to make $\mathcal{V}$ accept ($\mathcal{P}$ is no longer available at this stage). In an *active* attack, the adversary has the ability to interact with $\mathcal{P}$ in phase one, while phase two remains the same. In phase one of the strongest *man-in-the-middle* attack, the adversary can arbitrarily interact with $\mathcal{P}$ and $\mathcal{V}$ (polynomially many concurrent executions are allowed). The second phase is the same as before.

The most classical uses of authentication protocols are “handshakes” between two parties, who wish to communicate over a network, or access control - every time you use an access card to enter a building, the card and the reader are most probably running an authentication protocol.

Lightweight Authentication Computationally weak devices, such as RFID tags and some contactless smart cards, are playing an important role nowadays. Their widespread use is increasing the chance of their misuse, putting high demand on secure protocols and constructions fit for these devices. Very often, block-ciphers are the golden hammer for building MACs, as the communicating parties have highly optimized implementations of the main building block (such as AES co-processor on modern processors). However, weak devices do not have the circuitry capable of performing operations required by MACs, most often for computing conventional block-ciphers (AES). One of the solutions could be a dedicated cipher for weak hardware, such as PRESENT [15]. Additionally, a new way of building lightweight authentication protocols is to construct them directly from cryptographic assumptions “friendly” to weak hardware. Sampling random bits is considered computationally expensive on regular computers and could be the bottleneck of any construction. On the other hand, this is not true on weak devices, making computational problems based in some way on randomness, like Learning Parity with Noise (LPN), more interesting. In addition, if the computation performed during the protocol based on such assumptions is within the capabilities of this weak hardware, then such schemes are clearly the more suitable candidates for cryptography on weak devices.

Authenticated Encryption Intuitively, when we exchange messages, we want them to be unreadable (encrypted), with guaranteed integrity, and authenticated (with known origin). Most of the times a message is encrypted first and then a MAC is attached to it to achieve all three goals. However, instead of encrypt-then-mac, one can merge these two operations together into one construction, resulting in a more efficient scheme with the same level of security. Such primitive is called authenticated encryption and we talk about it more in Section 3.5.

1.4 Exact Bounds

In this section we explain the importance of exact security bounds and why they are so useful. When we look at the security of a cryptographic scheme, we consider the advantage of any adversary in a distinguishing experiment. The goal of the adversary in such experiment is to distinguish a cryptographic scheme from some ideal object whose behaviour this scheme tries to mimic. We can see a simple depiction of its various levels

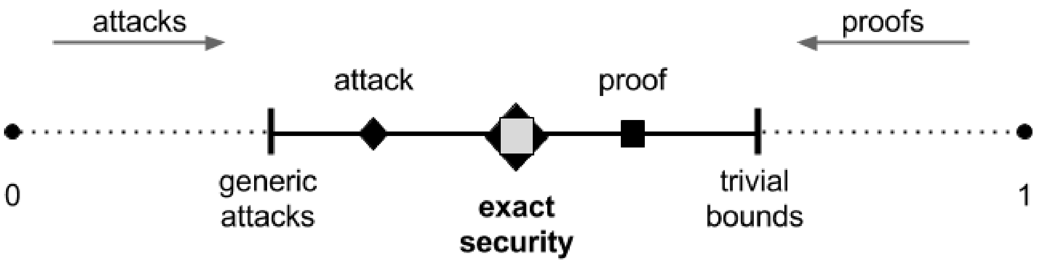


Figure 1.1: Distinguishing advantage.

in Figure 1.1 (as the advantage is just a probability measure, the most basic bounds are 0 and 1). There are two important values called the lower and upper bound. There are two ways to place them in the graph, or change their values; the first way is to come up with a new proof of security and move the right (upper) bound towards the left (we show that no attacker can achieve a higher distinguishing advantage than this value, hence we improve the security guarantees for the given scheme). The second way is to move the left (lower) bound by showing an attack on the construction (we show that there is an attacker that achieves this distinguishing advantage, hence the security guarantees for this scheme cannot be better than this value). If the lower bound is not equal to the upper, we say there is a tightness gap. Ideally, we would like the lower and upper bound to meet in order to know the exact security of the scheme. Additionally, the more this exact security bound goes towards the left on the graph, the better.

There are two additional points on the picture. The first one is called generic attacks - it is a bound that can be reached by using attack that is build for a class of cryptographic constructions with the same basic structure. They give us a hint that the security of some schemes has a natural limit. The second point we would like to mention is named trivial bounds. It marks a basic level of security we expect from a modern cryptographic scheme.

1.5 Outline and contributions of this thesis

Apart from the introduction you have seen above, this thesis will roughly consist of the following sections. First, in Chapter 2 we will describe the general notation that we will use throughout Chapters 3-5. More notation will be added in Chapters 6, and 7, which is specific to those chapters.

Next, we shall explore what Message Authentication Codes are into greater depth in Chapter 3. We will introduce security notions, different applications and the currently most widely used schemes.

In Chapter 4 we move on to describe the first part of the actual contributions of this thesis. We give a more accessible overview of the work summarized in the paper titled "The Exact PRF-Security of NMAC and HMAC", which was published at Crypto 2014 conference [24]. Briefly, NMAC is a mode of operation which turns a fixed input-length keyed hash function f into a variable input-length function. A practical single-key variant of NMAC called HMAC is a very popular and widely deployed message authentication code (MAC). Our first contribution is a simpler and *uniform* security proof for NMAC. Our proof is based on a previous result by Bellare et al., who show that cascading is a good domain extension for PRFs when restricted to prefix-free queries. We then show that the proved bound is basically tight by showing an attack. Finally, we analyze the PRF-security of a modification of NMAC called NI [4] that differs mainly by using a compression function with an additional keying input. We carry out an information theoretic analysis, obtaining a tight bound for this step. The proof uses some combinatorial techniques originally developed for proving the security of CBC-MAC (see Section 3.2).

The second contribution of this thesis is summarized in Chapter 5, which describes a paper titled "The Exact Security of PMAC" published in Transactions on Symmetric Cryptology journal and presented at Fast Software Encryption 2017 conference [26]. We explore the exact security of another MAC, called PMAC (Parallelizable MAC). PMAC is a simple and parallel block-cipher mode of operation, which was introduced by Black and Rogaway at Eurocrypt 2002. If instantiated with a (pseudo)random permutation over n-bit strings, PMAC constitutes a provably secure variable input-length (pseudo)random function. In this work we show that the currently best known security bound is tight by giving an attack with the same advantage. We also give a more general version of the attack that also applies to other variations of PMAC. Moreover, we investigate if the security of PMAC can be further improved by some slight modifications in its design. We prove that the security can increase in certain cases, matching a security upper bound set by simple extension attacks.

Lastly, Chapter 6 contains the full version of the "The Exact PRF-Security of NMAC and HMAC" paper [24], while in Chapter 7 you find "The Exact Security of PMAC" paper [26].

2 Preliminaries

Basic Definitions. We reserve the letter λ do denote the empty string. For $n \in \mathbb{N}$ we define $[n] := \{1, \dots, n\}$, and $\{0,1\}^{n*} := \bigcup_{z \in \mathbb{N}} \{0,1\}^{nz}$ denotes the set of all bitstrings whose length is a multiple of n. In a slight abuse of notation, we interchangeably view strings from $\{0,1\}^{n*}$ also as finite sequences of blocks from $\{0,1\}^n$, i.e., for $s \in \{0,1\}^{nz}$ we also write $s = (s_1, \dots, s_z)$ for $s_i \in \{0,1\}^n$. The (bit)length of a string w is $|w|$, and if $|w|$ is a multiple of n, $|w|_n = |w|/n$ denotes the length in n bit blocks. $w^\ell := w \| w \| \dots \| w$ denotes the ℓ-fold concatenation of w. We usually denote sets by calligraphic letters like $\mathcal{X}$. $\mathcal{F}_{b,c}$ (resp. $\mathcal{F}_{b*,c}$) denotes the set of all functions from $\{0,1\}^b$ to $\{0,1\}^c$ (resp. from $\{0,1\}^{b*}$ to $\{0,1\}^c$), $\mathcal{F}_{\mathbb{N},b}$ is the set of all functions $\mathbb{N} \to \{0,1\}^b$ and $\mathcal{P}_n$ the set of all permutations on $\{0,1\}^n$. If P is a (finite or infinite) sequence, then by $P_{[\ell]}$ we denote a tuple containing the first ℓ elements of P. A *partition* of a set S is a collection of non-empty subsets A_i, such that if $A_i \neq A_j$, then $A_i \bigcap A_j = \emptyset$, and $\bigcup A_i = S$. We use the symbol $\oplus$ to denote the bit-wise XOR of two bitstrings of the same length. When talking about a graph G, we denote its set of vertices and edges by $V(\mathsf{G})$ and $E(\mathsf{G})$ respectively. For an integer n, $d(n) = |\{i \in \mathbb{N} : i \mid n\}|$ is the number of its positive divisors and

$$d'(n) := \max_{n' \in \{1,\dots,n\}} |\{d \in \mathbb{N} : d \mid n'\}| \approx n^{1/\ln \ln n}$$

is the maximum, over all positive integers $n' \leq n$, of the number of positive divisors of n'. More precisely, we have $\forall \varepsilon > 0 \; \exists n_0 \; \forall n > n_0 \colon d(n) < n^{(1+\varepsilon)/\ln \ln n}$ [29].

Multisets. We denote with $\mathsf{mult}(x, \mathcal{X})$ the multiplicity of an element x in a multiset $\mathcal{X}$. $\mathcal{X}^{\downarrow}$ is the subset of $\mathcal{X}$ that contains only the elements of odd multiplicity, i.e.,

$$\mathcal{X}^{\downarrow} = \{x \in \mathcal{X} : \mathsf{mult}(x, \mathcal{X}) \mod 2 = 1\} .$$

Groups and Cosets. For a definition of a commutative group and a discussion of the notions introduced below, see e.g. [33]. All the groups that we consider in this paper will be commutative, and we will use additive notation for groups. A *subgroup* of G is any subset H that is a group by itself. The *order* of G, denoted $|G|$ is the number of its elements. Lagrange's theorem states that if H is a subgroup of G, then $|H|$ divides $|G|$.

Let G be a group, and H its subgroup. Take $g \in G$. Then the set $g + H := \{g + h : h \in H\}$ is called a *coset of H in G*. Note that trivially any group G is a coset (of G in G), we call a coset *proper* if it is not a group. The set of different cosets of H in G forms a partition of G; and moreover, H itself appears in it as the coset $0 + H$, where 0 is the neutral element of G (and H). The size of a coset is again referred to as its *order*. Finally, the order of G is equal to the product of the order of H and the number of different cosets of H.

Random Variables and Experiments. Random variables and concrete values they can take are usually denoted by upper-case letters $X, Y, \ldots$, and lower-case letters $x, y, \ldots$ respectively.

If $\mathcal{M}$ is a distribution (respectively, a set), then we denote by $X \xleftarrow{\$} \mathcal{M}$ sampling the random variable X according to $\mathcal{M}$ (respectively, choosing it uniformly at random from $\mathcal{M}$). By X^ℓ we denote ℓ independent and identically distributed copies of a random variable X. A joint probability distribution of q random variables $(X_1, \ldots, X_q)$ is *k-wise independent*, if its restriction to any k coordinates is uniform over its domain, e.g., if all X_i have domain $\{0,1\}^n$

$$\forall i_1, \ldots, i_k\ ,\ 1 \leq i_1 < \cdots < i_k \leq q\ ;\ \forall x_1, \ldots, x_k \in \{0,1\}^n\ :$$
$$\mathsf{Pr}_{(X_1,\ldots,X_q)}\left((X_{i_1}, \ldots, X_{i_k}) = (x_1, \ldots, x_k)\right) = \left(2^{-n}\right)^k\ .$$

More generally, let $\mathcal{M}_n$ be a probability distribution over $\mathcal{F}_{\mathbb{N},n}$. In this case, we call $\mathcal{M}_n$ *k-wise independent*, if any k outputs of $f(\cdot)$ sampled from $\mathcal{M}_n$ are independent. Formally, $\mathcal{M}_n$ is k-wise independent, if:

$$\forall i_1, \ldots, i_k\ ,\ 1 \leq i_1 < \cdots < i_k\ ;\ \forall x_1, \ldots, x_k \in \{0,1\}^n\ :$$
$$\mathop{\mathsf{Pr}}_{f \leftarrow \mathcal{M}_n}\Big(\left(f(i_1), \ldots, f(i_k)\right) = (x_1, \ldots, x_k)\Big) = \left(2^{-n}\right)^k\ .$$

Adversaries. In this thesis an adversary is a probabilistic (running in polynomial time, or computationally unbounded) algorithm, sometimes with access to an oracle $\mathcal{O}(\cdot)$. We use sans-serif letters for adversaries, e.g., $\mathsf{A}^{\mathcal{O}(\cdot)}$, and will only consider "distinguishers", which are adversaries, whose final output is just one bit. We distinguish adaptive and non-adaptive ones. Non-adaptive adversaries must submit all their queries before the start of an experiment, while the former can use information gained from previous queries to calculate the next query.

Pseudorandom functions and permutations. Here we give the definitions of a Pseudorandom function (PRF), and of a Pseudorandom permutation (PRP). If the first component of the input to a function f is to be seen as a key, we sometimes call f a *keyed* function, where the first part of the input is referred to as the key (and $\mathcal{K}$ being called the *keyspace* of f). For a keyed function $f\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ under a key $k \in \mathcal{K}$ we often write $f_k(\cdot)$ instead of $f(k, \cdot)$. Given a variable input-length keyed function $f\colon \mathcal{K} \times \{0,1\}^{n*} \to \{0,1\}^n$, the PRF-advantage of an adversary A against f is defined as

$$\mathbf{Adv}_f^{\mathrm{prf}}(\mathsf{A}) := \mathsf{Pr}[K \leftarrow \mathcal{K}\ :\ \mathsf{A}^{f_K(\cdot)} = 1] - \mathsf{Pr}[\mathsf{R} \leftarrow \mathcal{F}_{n*,n}\ :\ \mathsf{A}^{\mathsf{R}(\cdot)} = 1]\ .$$

We also define

$$\mathbf{Adv}_f^{\mathrm{prf}}(q, \ell, t) := \max_{\mathsf{A}} \mathbf{Adv}_f^{\mathrm{prf}}(\mathsf{A})$$

where the maximum goes over all adversaries that run in time at most t, and ask at most q queries, each of length at most ℓ (in n-bit blocks). If we consider computationally unbounded adversaries, we drop the last argument, i.e., $\mathbf{Adv}_f^{\mathrm{prf}}(q, \ell) := \mathbf{Adv}_f^{\mathrm{prf}}(q, \ell, \infty)$.

Pseudorandom permutations (PRPs), and their security notions are defined analogously. Given a keyed permutation (i.e., a block-cipher) $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, the PRP-advantage of an adversary A against E is defined as

$$\mathbf{Adv}_E^{\mathrm{prp}}(\mathsf{A}) := \mathsf{Pr}[K \leftarrow \mathcal{K}\ :\ \mathsf{A}^{E_K(\cdot)} = 1] - \mathsf{Pr}[\mathsf{P} \leftarrow \mathcal{P}_n\ :\ \mathsf{A}^{\mathsf{P}(\cdot)} = 1]\ .$$

and

$$\mathbf{Adv}_E^{\mathrm{prp}}(q,t) := \max_{\mathsf{A}} \mathbf{Adv}_E^{\mathrm{prp}}(\mathsf{A})$$

where the maximum goes over all adversaries that run in time at most t and ask at most q queries.

The following quality-characterization of PRFs will be used in Chapters 4 and 6. A variable input-length keyed function $f : \{0,1\}^c \times \{0,1\}^{b*} \to \{0,1\}^c$ is an:

- $(\varepsilon, t, q, \ell)$*-secure PRF*, if we have $\mathbf{Adv}_f^{\mathrm{prf}}(q,\ell,t) \leq \varepsilon$.
- $(\varepsilon, t, q, \ell)$*-NA-secure PRF*, if the above is true for all adversaries A that choose their queries non-adaptively (i.e., A has to choose its q queries before seeing any of the outputs).
- $(\varepsilon, t, q, \ell)$*-PF-secure PRF*, if the above is true for all adversaries A that choose their queries to be prefix-free (i.e., no query is a prefix of another query).
- $(\varepsilon, t, q, \ell)$*-NA-PF-secure PRF*, if the above is true for all adversaries A that choose queries *both* non-adaptively and prefix-free.

For fixed input-length functions, we omit the parameter ℓ. Moreover, we refer to an adversary A as an $(\varepsilon, t, q, \ell)$-PRF adversary against f if it runs in time t, asks at most q queries each consisting of at most ℓ blocks, and achieves the advantage $\mathbf{Adv}_f^{\mathrm{prf}}(A) = \varepsilon$. We refer analogously to adversaries for the other PRF-notions defined above.

Message authentication codes. A message authentication code (MAC) is a pair of algorithms called a *tagging* and a *verification* algorithm. The tagging algorithm has two inputs - a key and a message, and one output - a *tag*. The verification algorithm has three inputs - a key, a message, and a tag. It outputs a bit representing the validity of the tag for the given message and key. The standard notion of security for MACs is *unforgeability under chosen-message attack*. It is well-known [9] that every PRF achieves this security property.

Collision security. For a keyed function $f \colon \mathcal{K} \times \{0,1\}^{n*} \to \{0,1\}$, we define

$$\mathbf{Adv}_f^{\mathrm{col}}(q,\ell) := \max_{M_1,\ldots,M_q} \mathsf{Pr}_{K \leftarrow \mathcal{K}} \left[\exists\, i \neq j \;:\; f_K(M_i) = f_K(M_j)\right] ,$$

where the maximum goes over all q tuples of distinct messages of length at most ℓ blocks. With regards to collisions, two useful claims were given in [46], see also [32] for the proof of claim (ii) and [45] for further discussion. Here we give a very simplified version of both of these claims. Firstly, a random system for our purposes is a composite object consisting of two chained functions - an inner function, and an outer random function. These functions are chained such that the output of the inner function serves as an input to the outer function. The output of the outer functions serves as an output for the whole composite object, while the input to the object is fed as input to the inner function.

Lemma 1. *Let* $\mathbf{F}$ *and* $\mathbf{G}$ *be random systems. Let* $\mathcal{A}$ *be a binary value indicating that we have seen no collision on the output of the inner function of* $\mathbf{F}$ *(*$\mathcal{A}$ *is true if no collision has been observed, false otherwise), let* $\mathbf{D}$ *be a distinguisher asking* q *queries. Then:*

(i) *[46, Lemma 7] If* $\mathbf{F}$ *is equivalent to* $\mathbf{G}$ *conditioned on* $\mathcal{A}$ *being true, then the advantage of* $\mathbf{D}$ *is less than or equal to the probability of making* $\mathcal{A}$ *false.*

(ii) *[46, Theorem 2] For the experiment above, choosing queries adaptively does not increase the success probability of* $\mathbf{D}$*.*

3 Message Authentication Codes

A Message Authentication Code is one of the basic cryptographic primitives that deals with the task of authentication. In common speech, the name MAC is often given to a short piece of information that is attached to a message to achieve two of the three basic cryptographic goals - integrity (the message has not been tampered with), and authenticity (message was sent by the honest party). In this thesis we use the proper name for this piece of information - a tag (or MAC-tag). As mentioned above, a MAC is a cryptographic primitive and its definition consists of three algorithms called the *key-generation*, the *tagging*, and the *verification* algorithm. The key-generation algorithm generates a key of some 'quality' (length) that is specified by its input, the security parameter. The tagging algorithm has two inputs - a key and a message, and one output - a *tag*. The verification algorithm has three inputs - a key, a message, and a tag. It outputs a bit representing the validity of the tag for the given message, and the key. A simple illustration of the tagging and verification algorithms can be seen in Figure 3.1. The textbook security notion for a MAC is called existential unforgeability under chosen message attack (uf-cma). It is well-known [9] that every PRF achieves this security property and we will use this fact later on.

Formal definitions Here, we summarize the paragraph above more formally. Following the definition from [35], we define a MAC as:

Definition 1 (Message authentication Code)**.** A message authentication code, MAC, consists of three probabilistic polynomial time algorithms $(\mathtt{Gen}, \mathtt{Mac}, \mathtt{Vrfy})$ such that:

1. The `key-generation algorithm Gen` takes as input the security parameter 1^n and outputs a key k with $|k| \geq n$.

2. The `tag-generation algorithm Mac` takes as input a key k and a message $m \in \{0,1\}^*$, and outputs a tag t. Since this algorithm may be randomized, we write this as $t \leftarrow \mathtt{Mac}_k(m)$.

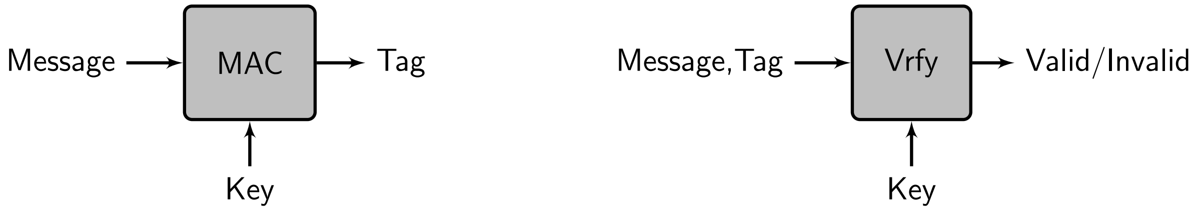


Figure 3.1: The tagging and verification algorithms.

3. The deterministic `verification algorithm Vrfy` takes as input a key k, a message m, and a tag t. It outputs a bit b, with $b = 1$ meaning `valid` and $b = 0$ meaning `invalid`. We write this as $b := \mathtt{Vrfy}_k(m, t)$.

It is required that for for every n, every key k output by $\mathtt{Gen}(1^n)$, and every $m \in \{0,1\}^*$, it holds that $\mathtt{Vrfy}_k(m, \mathtt{Mac}_k(m)) = 1$. If there is a function l, such that for every k output by $\mathtt{Gen}(1^n)$, algorithm $\mathtt{Mac}_k$ is only defined for messages $m \in \{0,1\}^{l(n)}$, then we call the scheme a `fixed-length MAC for messages of length` $l(n)$.

Note that even though we formally specified a verification algorithm, usually for deterministic MACs the verification algorithm just recomputes the tag and compares it to the one it received. In this sense, the tagging and verification algorithms will be the same.

When defining a security notion, we would like to capture all the different methods the adversary can interact with the honest parties. Apart from simply observing the communication, we also want to consider the possibility of the adversary influencing the content of the exchanged messages as well. Therefore, in our security model, we want to allow the adversary to see the tags computed under the valid key on messages of *their* choice. This is formally captured by the following textbook notion [35]. Consider the following experiment:

Definition (The message authentication experiment $\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n)$)**.** The experiment goes as follows:

1. A key k is generated by running $\mathsf{Gen}(1^n)$.

2. The adversary A is given input 1^n and oracle access to $\mathsf{Mac}_k(\cdot)$. The adversary eventually outputs (m, t). Let $\mathcal{Q}$ denote the set of all queries that A asked its oracle.

3. A `succeeds` if and only if (1) $\mathtt{Vrfy}_k(m, t) = 1$ and (2) $m \notin \mathcal{Q}$. In that case the output of the experiment is defined to be 1.

A MAC is secure if no efficient adversary can succeed in the above experiment with non-negligible probability:

Definition 2 (Existential unforgeablity under chosen message attack ($\mathsf{uf} - \mathsf{cma}$))**.** A message authentication code $\Pi = (\mathtt{Gen}, \mathtt{Mac}, \mathtt{Vrfy})$ is `existentially unforgeable under adaptive chosen-message attack`, or just `secure`, if for all probabilistic polynomial-time adversaries $\mathcal{A}$, there is a negligible function `negl`, such that:

$$\mathsf{Pr}[\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathtt{negl}(n).$$

Building MACs. The most natural way to build a MAC is to construct it from another cryptographic primitive, most prominently from a pseudorandom function. To be more precise, it can be easily shown that every PRF with a sufficiently large range *is* a MAC. It follows that a PRF is a strictly stronger primitive, candidate examples including the GGM scheme [28], or constructions based on AES (Advanced Encryption Standard). This relationship makes the task of constructing MACs easier, however, such straightforward constructions from candidate PRFs could be computationally expensive and provide unnecessary level of security for certain scenarios. A second approach to building MACs avoids going through a PRF, but the MAC is built directly from some cryptographic assumption (e.g. Computational Diffie-Hellman (CDH), or Decisional Diffie-Hellman (DDH) assumption [21]). These constructions are simpler, but building MACs from such assumptions has its caveats in large keys, or large amount of bits being transferred during the communication. Last, but not least, MACs can be also built from hash functions (SHA3, Secure Hash Algorithm 3), or modes of operation for block-ciphers (*e.g.* CBC-MAC, Cipher Block Chaining Message Authentication Code). Analysis of MACs based on hash functions and block-ciphers forms the core of this thesis and we will introduce them in more detail later in this chapter.

Connections to other primitives A very similar cryptographic primitive to a MAC is called the digital signature. A MAC is a symmetric primitive, hence the two parties that wish to exchange messages use the same (symmetric) key. A digital signature is an asymmetric primitive - the two parties have different (but mathematically related) keys. This translates into a simple narrative - whoever can verify a MAC can also compute it using the same secret key. On the other hand, only the entity in possession of the secret signing key can generate signatures, while anybody can verify these signatures using a publicly known verification key. Even though it seems that digital signatures are a more powerful primitive, their computation is much slower than the computation of a MAC, therefore MACs are still used whenever possible. As mentioned before, MACs are also connected to authentication protocols. We note that a secure MAC implies a man-in-the-middle secure 2-round authentication protocol (one party sends a challenge, while the other sends back the corresponding MAC) [36]. Additionally, as we mentioned in the previous paragraph, PRFs are closely connected to MACs as well - any secure PRF automatically constitutes a secure MAC [9]. Last, but not least, a *checksum* (or an error-detection code) is a non-cryptographic primitive used to check for integrity of messages, but *not* their authenticity. This means they are designed to protect against random errors (like electrical interference in wires), but not against malicious adversaries. On the other hand, the computation of checksums is much simpler and less expensive.

3.1 HMAC

HMAC (and its sister algorithm NMAC) is a deterministic MAC algorithm proposed by Bellare, Canetti and Krawczyk in 1996 [7]. A simple depiction of the algorithm is given in Figure 3.2. This popular algorithm is build from a cryptographic hash function and a secret key. In principle, any cryptographic hash function can be used for its computation, but HMAC-SHA1 and HMAC-MD5 are standardised [57], and used within IPsec and TLS protocols (based on SHA1 and MD5, accordingly). The security of HMAC is dependent on the cryptographic strength of the underlying hash function, the size of its hash output, and on the size of the key. NMAC differs from HMAC in the way they handle keys - NMAC is more suited for security proofs (that can be later lifted to HMAC), but HMAC is much easier to implement in practice and is used in real constructions. We postpone the precise introduction of the differences until Chapter 6 and will talk only about HMAC until then.

We can think of HMAC as an iterative hash function. An iterative hash function divides a message into separate blocks of a fixed size and then iteratively applies a compression function on these blocks, where the output of a previous iteration together with a fresh message block are used as inputs. The size of the output of HMAC is equivalent to

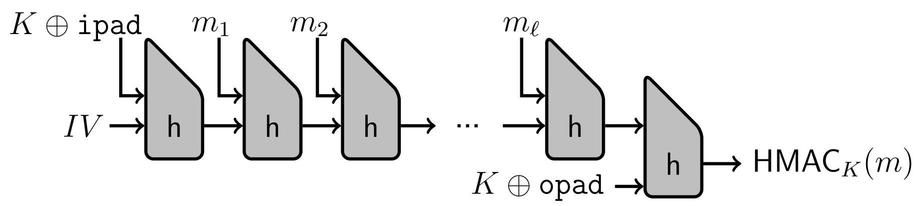


Figure 3.2: The construction HMAC_K.

the output size of the underlying hash function, unless truncated, which is also possible. HMAC tries to mimic the most intuitive way of building a MAC, which is simply prepending/appending a key to a message and hashing the result to produce a tag. When used naively, this method is, however, susceptible to a so called *length-extension attacks*. By using an additional call to a hash function at the end of the construction, HMAC achieves resistance to these attacks. Here, we would like to point a very similar construction that avoids these attacks slightly differently. The "sandwich" MAC (it both prepends *and* appends the key to a message) was proved to be secure as well [64].

Interestingly, the Keccak hash function, the winner of the recent SHA3 hash competition, is actually resistant against length-extension attacks (unlike SHA1, MD5, etc). Therefore, it doesn't need the nested approach of HMAC, and therefore can be used to generate a MAC by simply prepending the key to a message [1], hence saving one call to the underlying building block in terms of efficiency.

3.2 CBC-MAC

Together with HMAC, CBC-MAC is one of the most famous and widely deployed message authentication codes. You can see its sketch in Figure 3.3. Its design is borrowed from encryption, as CBC stands for Cipher-Block-Chaining, and it is a mode of operation for block-ciphers. However, it can be easily modified to form a MAC, hence CBC-MAC. The computation is almost the same as in encryption, apart from the initialization process and the fact that most of the ciphertext blocks are dropped, only the last one is output as a Tag. CBC-MAC has been standardised and is used, for example, in TLS. Apart from the pure CBC-MAC, there exist many variants of this algorithm, most prominently ECBC-MAC that has the Tag encrypted under a different key and benefits from a higher level of security.

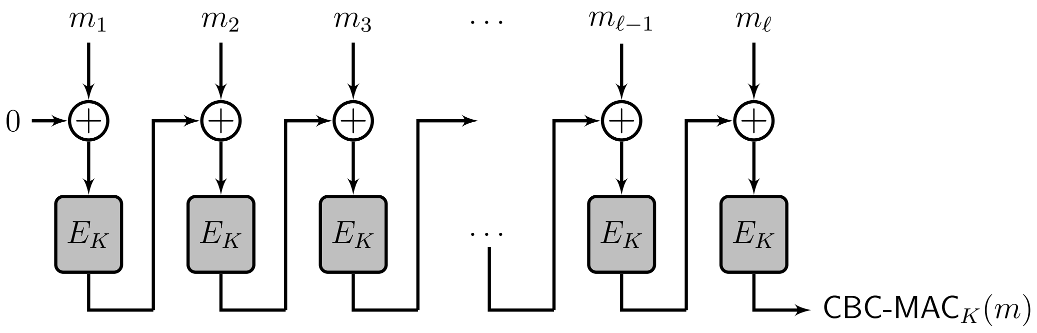


Figure 3.3: The construction $\mathsf{CBC\text{-}MAC}_K$.

Lastly, it is important to note that some combinatorial techniques developed for proving an improved bound for CBC [11] were used to prove the results of Chapters 4 and 6.

3.3 PMAC

PMAC is a parallelizable MAC that is based on a block-cipher, as opposed to being based on a hash-function, which was the case in HMAC. It was introduced by Black and Rogaway at Eurocrypt 2002 [14], and you can see its slightly simplified design in Figure 3.4. The security of PMAC is parametrized by the block-size and the quality of the underlying block-cipher over $\{0,1\}^n$, which is in turn dependable on the size of the secret key. The masks that are xor-ed to the message blocks are the multiplication of a fixed key-dependent value with a Canonical Gray code. The output size of PMAC is equal to the block-size of the block-cipher, with the possibility of truncation of the Tag. PMAC is slightly less efficient than, for example, modes based on CBC-MAC, but its main advantage is that unlike CBC-based MACs, it allows to process the message blocks fully in parallel. The parallel processing property was in fact the main goal behind its design. PMAC comes from the family of MAC algorithms of XCBC, OMAC, *etc.*, and is even vaguely connected to the authenticated cipher OCB. PMAC is not standardised and not as popular as HMAC, however, there is a pending patent on its design.

3.4 Others Macs

Probabilistic (or Lightweight) MACs Even though we talked about lightweight authentication in Section 1.3, here we would like to add more information directly connected to MACs. Until recently, MACs were purely deterministic. Recently, we have seen the introduction of probabilistic MACs, such as those based on LPN (learning parity with noise) assumption [36, 21, 44]. These MACs use randomness for the computation of the tags, hence the name probabilistic MACs. Consequently, this also means that for a fixed key, every message has a set of valid tags, not just a single tag (deterministic case) and the MAC has some nonzero completeness error. These MACs are especially suited for weak devices, such as RFID tags, as they replace hard computational operations with using more randomness, which is actually easy to get on this kind of devices. Additionally, they have also brought with them a need for new security definitions. As a response, existential

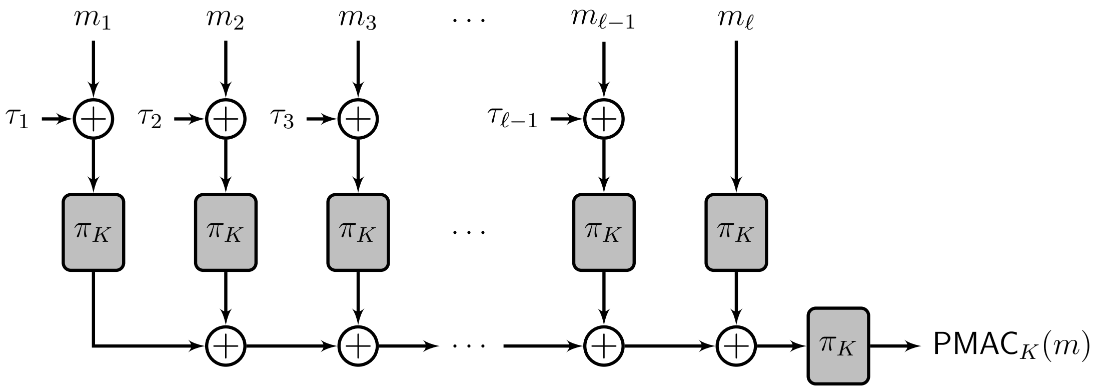


Figure 3.4: The construction PMAC_K.

unforgeability under chosen message with verification attack (uf-cmva) was introduced. The main idea is that for probabilistic MACs, the attacker is allowed to modify tags and send them to the verification oracle to check if it is valid. Note that for a deterministic MAC, there is only one valid MAC for every message, therefore, this notion is actually equivalent to uf-cma (Definition 2). As we deal neither with probabilistic MACs, nor with uf-cmva in this thesis, we point an interested reader to [21] for details.

Information-Theoretic MACs All the MACs we have talked about so far are secure only in the presence of *computationally bounded* adversaries. On the other hand, *information-theoretic* MACs (in fact, any information-theoretic primitive) are secure even in the presence of computationally unbounded ("all powerful") adversaries. Security in this sense means that the adversary cannot forge a valid tag with probability greater than taking a random guess, *i.e.* 2^{-n} for tags of size n-bits. We will not go into further detail, but information-theoretic security is achievable with some bounds on how many messages can be authenticated by honest parties per key. This number of messages is closely linked to the size of the keys, *e.g.* for a key of length $2n$, only a single message can be MAC-ed using a pairwise independent function to get 2^{-n} security. Similarly, for keys of length $(q+1)n$, we can compute a MAC for q messages using a q-wise independent function.

Wegman-Carter MACs A very influential class of MACs was proposed by Wegman and Carter in 1981 [63]. Their idea is to use so called universal hash-function families to efficiently hash down the message, and then encrypt it using a short key. One of the main constructions built on top of these ideas is UMAC, originally published at Crypto in 1999 by Black *et al.*[13].

3.5 Authenticated Encryption

In Section 1.1 we have talked about the CIA (confidentiality, integrity, authenticity) principle. In this Chapter, we talk about MACs - a primitive that is defined to bring us the latter two goals. The first goal, confidentiality, is generally achieved through encryption. By using encryption together with a MAC, we could achieve all three of the cryptographic goals. In most books you will find the Encrypt-then-MAC recipe to combine these two primitives. However, in the past decade a new primitive was proposed - authenticated encryption. Authenticated encryption, as the name suggests, aims to combine MACs with encryption to provide the most efficient primitive for achieving the CIA goals together. It is an encryption scheme that apart from the ciphertext also outputs a Tag that satisfies Definition 2 with slight modifications. Therefore, the output of the authenticated cipher, ciphertext and the tag, guarantee confidentiality, integrity, and authenticity at the same time. Several methods have been proposed, out of which OCB, mode of operation for block-ciphers ("offset codebook") [59], is the most popular. However, it suffers from some drawbacks - it is patented and therefore not freely available (like AES, for example), and there are already nontrivial attacks against it [23]. In January 2013 a new cryptographic competition called CAESAR [1] was announced, its goal being to select the best candidate for a standardised, patent free authenticated cipher. The submission deadline was in March 2014 and the competition is currently in its third round of selection.

[1] http://competitions.cr.yp.to/index.html

4 Exact Security of HMAC

We use this Chapter to revisit the HMAC algorithm and give an exposition of results shown in full detail in Chapter 6. On top of that, we give some additional details and look into some further research that was done after [24] was published.

To start with, let us briefly reintroduce the NMAC and HMAC algorithms. They were both proposed by Bellare, Canetti and Krawczyk in 1996 [7], and later standardized [39]. As we already mentioned in Chapter 3, HMAC has become very popular and widely used, *e.g.* in TLS. Although originally designed as a MAC, it is also often employed more broadly as a pseudorandom function (PRF). This is the case, for example, when used for key-derivation in TLS and IKE (the Internet Key Exchange protocol of IPsec). This proliferation into practice motivates the need for a good understanding of the exact security guarantees provided by NMAC and HMAC when used as a PRF.

Formally, NMAC is a mode of operation which transforms a keyed fixed input-length function $\mathsf{f} : \{0,1\}^c \times \{0,1\}^b \rightarrow \{0,1\}^c$ (with $b \geq c$) into a keyed variable input-length function $\mathsf{NMAC}^{\mathsf{f}} : \{0,1\}^{2c} \times \{0,1\}^{b*} \rightarrow \{0,1\}^c$ (where $\{0,1\}^{b*}$ denotes all bit strings whose length is a multiple of b) as

$$\mathsf{NMAC}^{\mathsf{f}}((K_1,K_2),M) := \mathsf{f}(K_2, \mathsf{Casc}^{\mathsf{f}}(K_1,M)\|0^{b-c})$$

where $\mathsf{Casc}^{\mathsf{f}} : \{0,1\}^c \times \{0,1\}^{b*} \rightarrow \{0,1\}^c$ is the cascade (also known as Merkle-Damgård) construction

$$\mathsf{Casc}^{\mathsf{f}}(K_1, m_1\|\ldots\|m_\ell) := \mathsf{f}(\ldots\mathsf{f}(\mathsf{f}(K_1,m_1),m_2)\ldots m_\ell)\ .$$

HMAC is a variant of NMAC tweaked for applicability in practice. As security proofs for NMAC can typically be lifted to HMAC, it is usually sufficient to analyse the security of the cleaner NMAC construction. We will discuss this point further in the next paragraph. As opposed to NMAC, the two keys (K_1,K_2) in HMAC are derived from a single key $K \in \{0,1\}^b$ by xor-ing it with two fixed b-bit strings ipad and opad. In addition, the keys are not given through the key-input of the compression function f, but are prepended to the message instead. This allows for the usage of existing implementations of hash functions that contain a hard-coded initialization vector (IV). Formally:

$$\begin{aligned}\mathsf{HMAC}^{\mathsf{f}}(K,m) \quad &:= \quad \mathsf{Casc}^{\mathsf{f}}(\mathsf{IV}, K_2\|\mathsf{Casc}^{\mathsf{f}}(\mathsf{IV},K_1\|m)\|\mathsf{fpad}) \\ & \quad \text{where } (K_1,K_2) := (K \oplus \mathsf{ipad}, K \oplus \mathsf{opad})\end{aligned}$$

and fpad is a fixed $(b-c)$-bit padding not affecting the security analysis.

As we mentioned above, the proofs in this paper consider NMAC. There is a standard reduction of HMAC-to-NMAC PRF-security given by Bellare [5], albeit under some additional requirements on the underlying compression function f. Informally, one needs to assume that f is a PRF even when keyed through the b-bit data input, as opposed to being

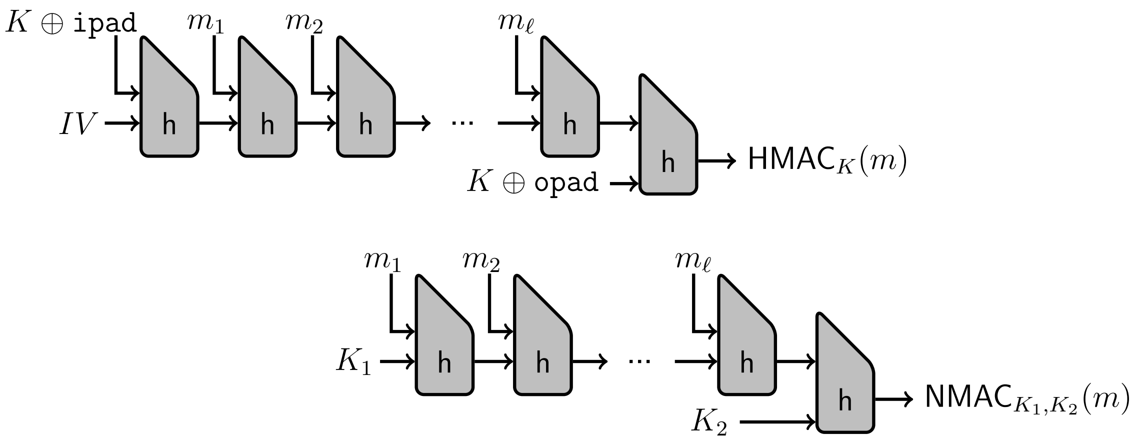


Figure 4.1: HMAC versus NMAC.

keyed by the c-bit chaining variable. Moreover, security of the single-key version of HMAC requires the PRF to be secure under a specific class of related-key attacks. Formally, the reductions are given in Lemmas 5.1 and 5.2 in the full version of [5] for the case of double- and single-keyed HMAC, respectively. Since these reductions only relate to NMAC via its PRF-security, they apply to our result in a blackbox way, thus giving clear statements also for HMAC.

We now discuss the results on PRF security of HMAC, which can be seen in Figure 4.2, depicted in the form of Section 1.4. Firstly, the lower bound of $q^2/2^n$ is due to generic birthday-bound attacks. Secondly, the original bound by Bellare *et al.* [7] prove that NMAC is a secure PRF under the condition that f is a PRF, and secondly that Casc$^{\mathsf{f}}$ is weakly collision-resistant (one requires that it is hard to find a pair of messages $M \neq M'$, such that Casc$^{\mathsf{f}}(K, M)$ = Casc$^{\mathsf{f}}(K, M')$ under a random key K). However, in HMAC instead of Casc$^{\mathsf{f}}$ we often use hash functions like MD5 or SHA-1, both of which have been found not to be weakly collision-resistant [61, 62]. Therefore, for these scenarios we cannot use the security proof by [7] we described above. We would like to point out that despite this fact, there are no know attacks (better than generic birthday-bound attacks) for NMAC or HMAC with MD5 or SHA-1.

Consequently, in a follow-up paper Bellare [5] tries to evaluate the PRF security without the weakly collision-resistant assumption, assuming only that the function f is a good PRF. The proved result roughly says that if f is an ε-secure PRF (against an adversary running in time t and asking q queries) and a γ-secure PRF (against time $O(\ell)$ and 2 queries), then NMAC$^{\mathsf{f}}$ is an $(\varepsilon + \ell q^2 \gamma)$-secure PRF against time t and q queries of length at most ℓ (in b-bit blocks). The most recent result on the PRF security of HMAC due to [38] claimed that HMAC is an $\varepsilon\ell$-secure PRF. However, this bound is falsified by an attack given in Section 4.2.

Our first contribution is a simpler, and as we will show, basically tight proof for the

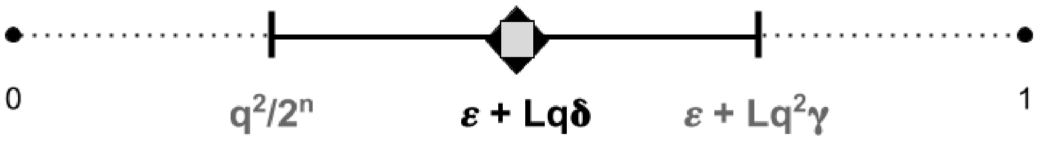


Figure 4.2: HMAC results overview.

PRF-security of NMAC^f assuming only that f is a PRF. The argument states that if f is an ε-secure PRF against q queries, then NMAC^f is roughly $\ell q \varepsilon$-secure against q queries of length at most ℓ blocks each. The actual result is more fine-grained, and expresses the security in terms of both the adaptive and non-adaptive security of f. We will discuss the details later in the following Section 4.1.

Following this result we also prove an upper bound for the PRF security of HMAC in Section 4.2, proving that the above lower bound is basically tight. From any PRF, we construct another PRF f for which NMAC^f can be broken with advantage $\Theta(\ell q \delta)$. This shows that our bound is tight for the practically most important case when $\ell q \delta$ is larger (or at least comparable) to ε. However, the above-mentioned attack achieving advantage $\ell q \delta$ against the NMAC^f construction for a particular f is not really satisfactory, as it assumes that f contains "pathological" features not expected from any natural PRF. This suggests that the bound obtained might be a too pessimistic approximation of the security level of NMAC one should expect in practice. A more "optimistic" way to capture the security of the NMAC construction itself is to assume that f behaves like an ideal function.

Consequently, as our second main positive result, we analyze the security of $\mathsf{NMAC}^{\mathbf{f}}$ in the information-theoretic setting where $\mathbf{f}$ is an ideal compression function. Not surprisingly, in this idealized setting we are able to give a much stronger $\ell d'(\ell) q^2/2^c$-bound where $d'(\ell) \approx \ell^{1/\ln \ln \ell}$ denotes the maximum number of divisors of any positive integer not greater than ℓ. This bound is of information-theoretic nature, hence being valid also for adversaries that are restricted by the number q and length ℓ of their queries, but are otherwise computationally unbounded. We also give matching attacks, showing that this bound is tight for constant q, and almost tight (i.e., up to the $d'(\ell)$ factor) for general q. The proof borrows combinatorial techniques originally developed for proving the security of the CBC-MAC [11]. These techniques need considerable adaptations, as in [11] the round function for the CBC mode is a *permutation* with a *fixed* key, whereas for NMAC we consider a *function* which is constantly *rekeyed*.

This rekeying makes the proof approach typically applied to constructions that use a PRF f under a fixed random secret key, much harder to perform. There, the analysis starts by replacing the PRF with an ideal random function (introducing an error that is upper-bounded by the PRF-security of f) and proceeds by a fully information-theoretic argument (note that this is the approach we take when analyzing PMAC in Chapters 5 and 7, where we replace a keyed permutation by a random permutation). Nevertheless, we try to make such an analysis by investigating the PRF-security of the nested iterated (NI) construction introduced in [4]. The construction NI^h is very similar to NMAC^f, but is based on a compression function h that (compared to f) takes an additional k-bit input which is used for keying instead of the chaining input. Hence, in our analysis NI^h uses h under the same key throughout the whole cascade. The modified keying allows for the simple switching argument from PRF to a random function we mentioned above. Additionally, we focus on enhancing the information-theoretic analysis that follows this switch and prove an essentially tight $\ell q^2/2^c$ bound for this step, improving significantly over the trivial bound of $\ell^2 q^2/2^c$.

Lastly, we would also like to mention that there is also a recent line of work investigating generic attacks against iterated hash-based MACs [53, 40, 50, 54]. These works present various attacks against MACs (e.g. related-key attack, universal forgeries, state recovery) that do not exploit the inner structure and potential weaknesses of the compression function, instead they rely solely on the iterative structure of the MACs.

4.1 New proof of security

In this section we analyze the PRF security of $\mathsf{NMAC}^{\mathsf{f}}$ in terms of the PRF-security of the underlying function f.

Before analyzing the actual $\mathsf{NMAC}^{\mathsf{f}}$ construction, we first look at the PRF security of the inner cascade $\mathsf{Casc}^{\mathsf{f}}$ when it is queried on prefix-free messages. We adapt the result of Bellare *et al.* [8] by modifying it to obtain security against *non-adaptive* adversaries, assuming only *non-adaptive security* of the underlying compression function f. This will allow us to give a more fine-grained security bound for NMAC, as it will both in terms of adaptive and non-adaptive security.

Proposition 1 ($\mathsf{Casc}^{\mathsf{f}}$ as a NA-PF-PRF)**.** *Let* $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ *be a compression function. There exists an explicit reduction* T *(described in the proof) such that for any* $(\varepsilon', t', q, \ell)$*-NA-PF-PRF adversary* A *against* $\mathsf{Casc}^{\mathsf{f}}$*,* T^{A} *is an* $(\varepsilon_{\mathsf{na}}, t, q)$*-NA-PRF adversary against* f *such that*

$$\varepsilon' \le \ell q \varepsilon_{\mathsf{na}} \qquad \text{and} \qquad t = t' + \tilde{O}(\ell q)\ .$$

The full proof based on the random system framework is given in Appendix 6.5.1. It uses a hybrid argument with an intermediate construction, hence it is split into two parts. The intuition is that if we can break the security of the cascade, then we must be able to break the security of at least one inner block of the cascade. However, each block is a secure PRF f, hence the statement is proved by contradiction.

The main theorem of this section is presented below. It relates the adaptive PRF-security of the construction $\mathsf{NMAC}^{\mathsf{f}}$ to both the adaptive and non-adaptive PRF-security of f.

Theorem 1 ($\mathsf{NMAC}^{\mathsf{f}}$ as a PRF)**.** *If* $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ *is an* (ε, t, q)*-secure PRF and an* $(\varepsilon_{\mathsf{na}}, t, q)$*-NA-secure PRF, then* $\mathsf{NMAC}^{\mathsf{f}}$ *is an* $(\varepsilon', t', q, \ell)$*-secure PRF with*

$$\varepsilon' = \varepsilon + (\ell+1) q \varepsilon_{\mathsf{na}} + \frac{q^2}{2^c} \qquad \text{and} \qquad t = t' + \tilde{O}(\ell q)\ . \tag{4.1}$$

The reduction is uniform. Concretely, there exist explicit reductions T_1 *and* T_2 *(described in the proof) such that for any* $(\varepsilon', t', q, \ell)$*-PRF adversary* A *against* $\mathsf{NMAC}^{\mathsf{f}}$*,*

1. $\mathsf{T}_1^{\mathsf{A}}$ *is an* (ε, t, q)*-PRF adversary against* f*,*

2. $\mathsf{T}_2^{\mathsf{A}}$ *is an* $(\varepsilon_{\mathsf{na}}, t, q)$*-NA-PRF adversary against* f*,*

and their parameters satisfy equations (4.1).

The formal proof (again in the random system framework) can be found in Section 6.3. It basically says that if an adversary breaks the PRF-security of $\mathsf{NMAC}^{\mathsf{f}}$, then it must break either the NA-PRF-security of the cascade, or the PRF-security of the outer PRF f. As we know that both of these are secure, then so must be $\mathsf{NMAC}^{\mathsf{f}}$.

4.2 Tight security bound via a new attack

In this section, we argue that the PRF-security bound for $\mathsf{NMAC}^{\mathsf{f}}$ obtained in Theorem 3 is essentially tight. First, we show that the term $\ell q \varepsilon_{\mathsf{na}}$ is unavoidable (up to a constant factor) by constructing a particular compression function f, which is an $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF, yet there is a simple attack against the PRF-security of $\mathsf{NMAC}^{\mathsf{f}}$ achieving advantage roughly $\ell q \varepsilon_{\mathsf{na}}$.

Proposition 2. *Let b, c, ℓ be positive integers such that $b \geq c$, let $\varepsilon_{\mathsf{na}} \in (0,1)$, and moreover, assume that pseudo-random functions exist. Then there exists a function $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ and an adversary A against $\mathsf{NMAC}^{\mathsf{f}}$ such that for any q that satisfies $\varepsilon_{\mathsf{na}} = \omega(q^2 2^{-b}, 2^{-c})$, we have:*

- f *is $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF;*
- *the adversary A, when asking q queries of length ℓ blocks each, runs in time $\tilde{O}(\ell q)$ and achieves distinguishing advantage*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{NMAC}^{\mathsf{f}}_K}(\mathsf{A}) = \Theta(\ell q \varepsilon_{\mathsf{na}}) .$$

In particular, for any $\varepsilon \in o(\ell q \varepsilon_{\mathsf{na}})$, and some $t \in \tilde{O}(\ell q)$, $\mathsf{NMAC}^{\mathsf{f}}$ is not an $(\varepsilon, t, q, \ell)$-secure PRF.

The full proof can be found in Chapter 6, Appendix 6.5.2. However, here we give the intuition of how the attack works.

To start with, we have to construct an $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF f that behaves pseudorandomly except for a small $\varepsilon_{\mathsf{na}}/2$-fraction of the keys (note that this is in line with the definition of a PRF from Chapter 2). We denote the set of these keys by $\mathcal{K}$ and refer to them as the *weak keys*. Then, the PRF f behaves as follows: under any weak key k, $\mathsf{f}(k, \cdot)$ outputs some constant value $w \in \mathcal{K}$ irrespective of its input. For us it is enough to assume $w := 0$.

As a next step, let us plug in the function f into the NMAC construction to form $\mathsf{NMAC}^{\mathsf{f}}_{K=(K_1,K_2)}$. For the attack, we sample a random message M of length $\ell - 1$ blocks at random. Then, we set $M_1 = M \| x_1$ and $M_2 = M \| x_2$ for two distinct blocks $x_1, x_2 \in \{0,1\}^b$. Then if some of the $\ell - 1$ intermediate values in the evaluation of the inner function $\mathsf{Casc}^{\mathsf{f}}(K_1, M)$ is in $\mathcal{K}$, then all following intermediate values are 0 (because of the way we defined the function f, once a weak key is hit, then the following keys are always 0). In particular, we have $\mathsf{Casc}^{\mathsf{f}}(K_1, M_i) = 0$ for both $i \in \{1, 2\}$, and hence also $\mathsf{NMAC}^{\mathsf{f}}(K, M_1) = \mathsf{NMAC}^{\mathsf{f}}(K, M_2) = \mathsf{f}_{K_2}(0)$. This implies that it is much more likely to get a collision for a pair of messages as described above for $\mathsf{NMAC}^{\mathsf{f}}_K$ than for a random function $\mathbf{R}$. Our adversary A maximises its chances by simply choosing $q/2$ message pairs at random as described above, and it outputs 1 if it observes a collision for at least one of those pairs. There are $q/2$ message pairs, each of length ℓ, so we have a total of $\ell q/2$ possibilities to "hit" a weak key, each having probability $\varepsilon_{\mathsf{na}}$. We then apply the union bound to give us a total probability of $\Theta(\ell q \varepsilon_{\mathsf{na}})$ for observing a collision when querying $\mathsf{NMAC}^{\mathsf{f}}_K$. On the other hand, the probability of observing a colliding pair when querying a random function $\mathbf{R}$ is much less, being only $O(q/2^c)$. We subtract these two values to prove the proposition.

4.3 HMAC extension and its security

In this section we analyze the PRF-security of the constructions called NI^{h} and $\mathsf{NI2}^{\mathsf{h}}$ (NI stands for Nested Iterated) under the assumption that the keyed compression function h is a PRF (when keyed via its k-bit input). Now, let us introduce the nested iterated (NI) construction defined in [4]. For this, we consider a keyed compression function $\mathsf{h}\colon \{0,1\}^k \times \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$. When such h is used in a cascading construction, its c-bit and b-bit inputs are used for the chaining value and the next block, respectively. In contrast to the function f considered previously, h has an additional k-bit input that is used for keying. Formally, for such h we define the *nested iterated* construction $\mathsf{NI}^{\mathsf{h}}\colon (\{0,1\}^k)^2 \times \{0,1\}^{b*} \to \{0,1\}^c$ as

$$\mathsf{NI}^{\mathsf{h}}_{K_1,K_2}(m) := \mathsf{h}_{K_2}(\mathsf{Casc}_{\mathbf{0}}^{\mathsf{h}_{K_1}}(m), |m|)$$

where $\mathbf{0}$ denotes the all zero bitstring 0^c and $|m|$ is the length of m encoded in b-bit blocks. For a graphical description, see Figure 4.3.

For a detailed discussion of the relationship of NI to NMAC, see [4]. For completeness, we also consider the modified version of NI that replaces the message length $|m|$ in the last (outer) call of the compression function by the constant bitstring 0^b, we denote this variant as $\mathsf{NI2}$. Formally, we have

$$\mathsf{NI2}^{\mathsf{h}}_{K_1,K_2}(m) := \mathsf{h}_{K_2}(\mathsf{Casc}_{\mathbf{0}}^{\mathsf{h}_{K_1}}(m), 0^b) .$$

We can say that $\mathsf{NI2}^{\mathsf{h}}$ is the variant of NI^{h} that does not care about the message length. Recall that $d'(n)$ denotes the maximum, over all positive integers $n' \leq n$, of the number of positive divisors of n'; i.e., $d'(n) := \max_{n' \in \{1,\ldots,n\}} |\{d \in \mathbb{N} : d \mid n'\}| \approx n^{1/\ln\ln n}$.

Our main theorem in this section is as follows:

Theorem 2. *If* $\mathsf{h}\colon \{0,1\}^k \times \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ *is an* (ε_1, t, q)*-secure PRF and an* $(\varepsilon_2, t, \ell q)$*-secure PRF, then* NI^{h} *is an* $(\varepsilon', t', q, \ell)$*-secure PRF with*

$$\varepsilon' = \varepsilon_1 + \varepsilon_2 + \frac{q^2}{2^c} \cdot \left(\ell + \frac{64\ell^4}{2^c}\right) \qquad \text{and} \qquad t = t' + \tilde{O}(\ell q) ,$$

and $\mathsf{NI2}^{\mathsf{h}}$ *is an* $(\varepsilon'', t'', q, \ell)$*-secure PRF with*

$$\varepsilon'' = \varepsilon_1 + \varepsilon_2 + \frac{q^2}{2^c} \cdot \left(\ell \cdot d'(\ell) + \frac{64\ell^4}{2^c}\right) \qquad \text{and} \qquad t = t'' + \tilde{O}(\ell q) .$$

As before, the proof can be found in Chapter 6, Theorem 4, and here we provide the underlying ideas for the proof. Firstly, let us explain the relationship between NI^{h} and

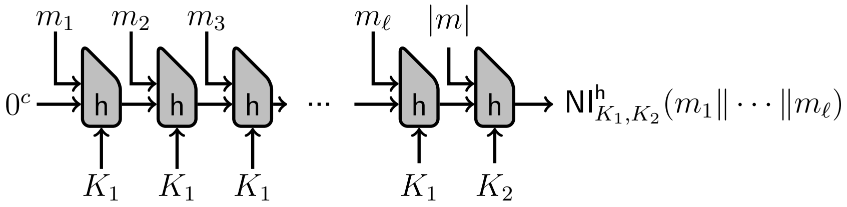


Figure 4.3: The construction $\mathsf{NI}^{\mathsf{h}}_{K_1,K_2}$.

$\mathsf{NI2}^{\mathsf{h}}$ in terms of proving the statement of the theorem. We start with the proof for $\mathsf{NI2}^{\mathsf{h}}$, as this is the "full" proof, where we work with messages of variable lenght. NI^{h} encodes the message length as part of the message, therefore, in the proof we have to work with messages of equal length (this fact will be clear later). Hence, the second proof is simpler, and can be derived from the first in a straightforward way (the set of messages with fixed length is a subset of the set of messages with variable length).

The proof for $\mathsf{NI2}^{\mathsf{h}}$ starts with the basic step of replacing the PRF h by an ideal compression function, and proceeding in the information-theoretic model (note that we have discussed this at the beginning of this chapter). This step adds the PRF-security bounds $(\varepsilon_1, \varepsilon_2)$ of h into the resulting bound for $\mathsf{NI2}^{\mathsf{h}}$.

In the second step of the proof, we observe that $\mathsf{NI2}$ with an ideal compression function behaves identically to a random function $\mathbf{R}$, as long as no non-trivial collision occurs in the outputs of the initial cascade. Let $\mathsf{CColl}(\ell)$ denote the probability that a random choice of the compression function $\mathbf{f}_1$ results in a collision in $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$, maximized over the choice of the two distinct inputs to the cascade m_1, m_2, consisting of at most ℓ blocks each. We then use the result of Maurer (see Lemma 1) to show that the advantage of the adversary is upper bounded by $q^2 \cdot \mathsf{CColl}(\ell)$. Thanks to $\mathbf{f}_2$, the responses of the composition $\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1}$ with $\mathbf{f}_2$ to distinct queries are clearly independent, uniformly random values and in this case the theorem says that distinguishing $\mathsf{NI2}$ is as hard as forcing a collision on the input to $\mathbf{f}_2$. If we then use the union bound for q^2 candidate message pairs, we arrive at the mentioned bound $q^2 \cdot \mathsf{CColl}(\ell)$.

Next, in the third part of the proof, we reduce estimating the probability of a collision on the output of the inner cascade of $\mathsf{NI2}$ with an ideal compression function to a counting problem of upper-bounding the number of graphs satisfying certain properties (modeling the computation of the cascade). We note that the probability $\mathsf{CColl}(\ell)$ could actually be trivially upper-bounded by $O(\ell^2/2^c)$ using a union-bound argument. Achieving a better and non-trivial bound on $\mathsf{CColl}(\ell)$ is actually the central part of our proof. In order to do that we use so called structure graphs, which represent the computation of $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$ on various inputs, and are inspired by [11].

To start with, let us assume two distinct messages m_1 and m_2 ($\mathcal{M} = (m_1, m_2)$) that we parse into b-bit blocks as $m_i = m_i^1 \| \cdots \| m_i^{\ell_i}$ for some $\ell_1, \ell_2 \leq \ell$. Then the *structure graph* $G_f^{\mathcal{M}}$ for these messages and a fixed compression function f is defined as the triple $G_f^{\mathcal{M}} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$, the sets of vertices, edges, and labels.

Intuitively, the vertices in $\mathcal{V}$ represent the evolution of the intermediate values of the chaining variable of $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$, while the directed edges connect consequtive computation steps. The edges are labelled with message blocks (one or more) that were used in the computation. If all the values of the chaining variable are distinct, $G_f^{\mathcal{M}}$ simply consists of two directed paths starting in the root vertex 0, representing the evaluation of $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$ on the messages m_1 and m_2 (the edges from $\mathcal{E}$ are labeled by the corresponding blocks). If some collisions among the values of the chaining variable occur, one can obtain the graph $G_f^{\mathcal{M}}$ by collapsing every pair of vertices corresponding to such collision into one vertex, as well as merging the edge labels in the natural way. Let $\mathcal{G}(\mathcal{M}) := \{G_f^{\mathcal{M}} : f \in \mathcal{F}(c+b, c)\}$ denote the set of all structure graphs associated with the message pair $\mathcal{M}$. For a fixed structure graph $G = G_f^{\mathcal{M}}$ we denote by $G_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{L}_i)$ the graph that is obtained after processing only the first i out of $\ell_1 + \ell_2$ blocks of $\mathcal{M}$. The idea is that we reveal the structure graph G step by step, i.e., by a sequence of transitions from G_{i-1} to G_i. The index i belongs to $\mathsf{fColl}(G)$ (and we say that the i-th step caused an f-collision), if during

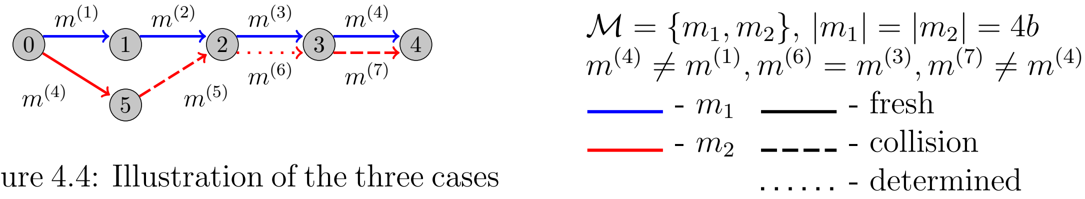


Figure 4.4: Illustration of the three cases from Lemma 2.

this computational step, instead of adding a new vertex to G_i, we arrive at a vertex already visited, while not following an existing edge already labeled with $m^{(i)}$ (i.e., not repeating a step we have made before).

To actually prove our result, first we upper-bound the probability of $G_F^{\mathcal{M}}$ taking the form of any particular fixed structure graph $g \in \mathcal{G}(\mathcal{M})$.

Lemma 2. *Let $F \leftarrow \mathcal{F}(c+b, c)$ be chosen uniformly at random. For a fixed graph $g \in \mathcal{G}(\mathcal{M})$ we have*

$$\mathsf{Pr}^F\left[G_F^{\mathcal{M}} = g\right] \leq 2^{-c \cdot |\mathsf{fColl}(g)|} .$$

The full argument can be found in the proof of Lemma 7. The high level idea is as follows. We pick any graph g and then follow $G_F^{\mathcal{M}}$, revealing it step by step. Then we categorize each new edge as (also depicted in Figure 4.4)

Fresh: It arrives at a new vertex not present in g_i.

Determined: It follows an already existing edge.

Collision: It causes an f-collision In this case, G_{i+1} will stay consistent with g if and only if its $(i+1)$-th edge lands on precisely the same vertex as in g_{i+1}. The probability of this event is 2^{-c}, as the $i+1$-th chaining variable is uniformly random over $\{0,1\}^n$ and not determined in the first i steps.

Since the third case occurs exactly $|\mathsf{fColl}(g)|$ times, if we trivially upper-bound the probabilities in the other two cases by 1, we obtain the final bound $\mathsf{Pr}[G = g] \leq 2^{-c \cdot |\mathsf{fColl}(g)|}$. Using this Lemma 2, it is easy to see that the event that at least two f-collisions occur in G is highly unlikely (for a proof, see Lemma 8 in Section 6.4).

Lemma 3. *Let $F \leftarrow \mathcal{F}(c+b, c)$ be chosen uniformly at random. Then*

$$\mathsf{Pr}^F\left[\left|\mathsf{fColl}\left(G_F^{\mathcal{M}}\right)\right| \geq 2\right] \leq \frac{4(\ell_1 + \ell_2)^4}{2^{2c}} .$$

Finally, we give a bound on the number of the structure graphs with a single collision, and hence we conclude the proof argument. We do that by upper-bounding the value $\mathsf{CColl}(\ell)$. Let $\mathcal{M} := (m_1, m_2)$ be the two distinct messages of length at most ℓ blocks that maximize the probability $\mathsf{CColl}(\ell) := \max_{m_1 \neq m_2} \mathsf{Pr}^F\left[\mathsf{Casc}_{\mathbf{0}}^F(m_1) = \mathsf{Casc}_{\mathbf{0}}^F(m_2)\right]$. Then we claim the following about the structure graph corresponding to a collision for these two messages - their corresponding paths need to divert after some vertex and come back together again in a different vertex at least once. The second step corresponds to a collision, and by Lemma 3, we know that seeing two, or more, collisions is highly unlikely. The reason behind this claim is that the messages need to differ in at least one block (the diverting of paths), but if they are to collide, by the definition of structure graphs, they need to end in the same vertex (hence, the reunion part).

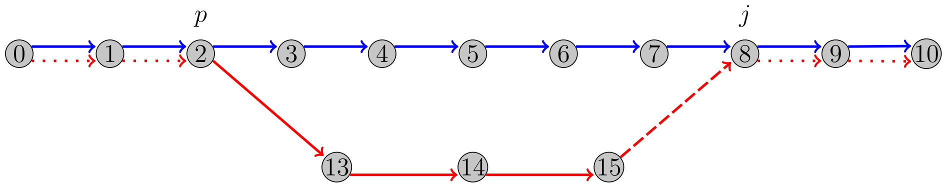


Figure 4.5: A sample graph from the set $\mathcal{H}_1$ in the proof of Lemma 4, with $p = 2$ and $j^* = 16$.

Therefore, in the last step of the proof, we count the number of graphs, where the paths corresponding to m_1, m_2 end in the same vertex, while causing precisely one collision. We denote these graph by $\mathcal{H}(\mathcal{M})$.

Recall that $d'(n)$ denotes the maximum, over all positive integers $n' \leq n$, of the number of positive divisors of n'; i.e., $d'(n) := \max_{n' \in \{1,\ldots,n\}} |\{d \in \mathbb{N} : d \mid n'\}|$.

Lemma 4. *For two distinct messages $\mathcal{M} = \{m_1, m_2\}$ each of length at most ℓ blocks we have $|\mathcal{H}(\mathcal{M})| \leq \ell d'(\ell)$. If the messages in $\mathcal{M}$ are of the same length then we have $|\mathcal{H}(\mathcal{M})| \leq \ell$.*

The proof from Section 6.4 handles the general case where we allow the messages m_1 and m_2 to have different lengths ℓ_1 and ℓ_2. Without loss of generality, let us assume that $\ell_1 \geq \ell_2$. Then we consider two scenarios - in the first one, the structure graphs contains a loop (or more), while in the second scenario it does not.

You can see a scenario without a loop in Figure 4.5. The intuition is that the two paths must part at some vertex p and rejoin at some further vertex j. However, this point j is determined by the length of the two messages - if at the time the paths rejoin the number of remaining vertices to travel is different for the two messages, then they will not end in the same vertex (note that the only way to "spent" redundant blocks is via another collision, which is unlikely, or via a loop, which we will cover later). This allows us to bound the number of structure graphs in this scenario by ℓ, the maximum number of different values of p.

Now, let us consider a scenario with a loop in it (note that two loops are created by two collisions, hence we assume just one - but we can reuse the loop multiple times). We split the graph into 3 different parts - part x is the common prefix of the messages, part y is the loop, while part z is the common suffix. You can find an illustration in Figure 4.6. We note that the length of the loop must divide the difference in message length of the two messages - imagine that the loop is a trick how to get rid of redundant message blocks, so both message paths end in the same block. Then, we observe that there are at most ℓ points that determine the start/end of the loop, as well as at most $d'(\ell)$ different loop lengths, such that the loop starts and ends in the same vertex. This allows us to bound the number of consistent structure graphs by $\ell \cdot d'(\ell)$.

The case of NI is almost exactly the same as $\mathsf{NI2}$, but we require the messages m_1 and m_2 in the definition of $\mathsf{CColl}(\ell)$ to be of the same length. This leads to the use of the second part of Lemma 4 that assumes equal-length messages, arriving at the claimed bound.

In Appendix 6.5.3 we show that Lemma 4 is actually tight. As a consequence, the adversary can simply choose q messages $m_1, \ldots, m_q$ of the form $m_i = x_i \| 0^{b(\ell-1)}$ for arbitrary distinct x_i's. In the proof we show that he succeeds with probability $\Omega(\ell q^2 / 2^c)$, hence giving the tight bound.

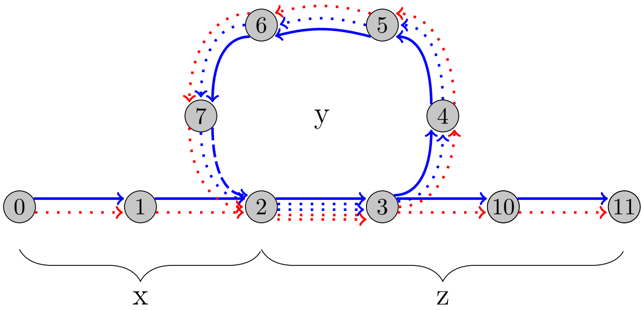


Figure 4.6: A sample graph from the set $\mathcal{H}_2$ in the proof of Lemma 4, with $i^* = 2$ and $j^* = 8$.

4.4 Further research

The paper on HMAC was followed by further research into this scheme. It is worth mentioning the paper by Bellare and Lysyanskaya [10], who study so called "Dual PRFs". When one wants to lift the security results of NMAC to results about HMAC, one of the assumptions that needs to be made is that the underlying compression function is secure, even if the roles of its two inputs (message, key) are reversed - meaning the compression function is a dual PRF. The authors of the paper discuss the notion of dual PRFs, possible constructions as well as constructions that use them, such as HMAC. Such research gives us further assurance about the close connection between NMAC and HMAC and strengthens the results of Chapter 6.

AMAC is a simple and fast candidate construction of a PRF from an MD-style hash function which applies the keyed hash function and then, unlike HMAC, an un-keyed output transform such as truncation. It is also a part of widely-deployed Ed25519 signature scheme. Its PRF security was also recently analyzed by [6].

In [25] the authors also analyze the PRF security of NMAC/HMAC by slightly modifying them. Apart from tight bounds on the PRF security, this modification protects both constructions against the recent generic attacks we mentioned previously [53, 40, 54].

Another open question lies within the way we model the adversary, as is is only allowed to access the ideal compression function $\mathbf{f}$ via its access to $\mathsf{NMAC}^{\mathbf{f}}$. Although this is in line with almost all previous works analyzing modes of operation in an idealized model, one has to be more careful to make arguments about the real world security for NMAC, than for example in the paper on the CBC-MAC [11]. The reason is that in the CBC mode, the underlying function (which is an ideal permutation) is used with a fixed key, and never re-keyed. Thus, to get a bound on the real-world PRF-security of the CBC-MAC when instantiated with a block cipher, one simply has to add (to the security bound in the ideal model) a term bounding the security of the block-cipher as a pseudorandom permutation (i.e., the advantage of an adversary in distinguishing the block-cipher, instantiated with a random key, from a random permutation).

The case of NMAC is more delicate, as here the inner cascade construction is constantly re-keying the ideal compression function $\mathbf{f}$. An ideal model that would capture security better, while still avoiding the poor bounds we get in the real model due to the existence of PRFs with an unrealistic pathological behavior, is to give the adversary not only access to $\mathsf{NMAC}^{\mathbf{f}}$, but also to the ideal $\mathbf{f} : \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$. Finding tight bounds in this model seems considerably challenging open problem.

5 Exact Security of PMAC

In this chapter we explore the exact security of PMAC, which we introduced in Section 3.3. The results described below are a simplified and more accessible version of Chapter 7, with some additional details and commentary.

As a remainder, PMAC is the most prominent parallelizable MAC, and is based on a block-cipher. It was invented by Jon Black and Phil Rogaway, and introduced at Eurocrypt 2002 [14]. To revisit its design, see Figure 5.2. The formal definition of PMAC looks as follows: the secret key consists of two permutations π, π' over $\{0,1\}^n$, and a function $\tau : \mathbb{N} \to \{0,1\}^n$ that determines the masks used in the construction. The output, when the input is a message $M = m_1 \| \dots \| m_\ell, m_i \in \{0,1\}^n$, is computed as

$$\mathsf{PMAC}_{\pi,\pi',\tau}(M) = \pi'\left(\bigoplus_{i=1}^{\ell} \pi(m_i \oplus \tau(i))\right) . \tag{5.1}$$

Additionally, the masks are computed as $\tau(i) = \gamma_i \cdot L$, where γ_i is the i-th Gray codeword (defined later). Our definition is slightly idealized, but we postpone the details until Section 5.1. The security of PMAC is usually analyzed assuming it is a pseudorandom function (defined in Chapter 2), and this is also the approach we take. Additionally, we work in the random permutation model - it means we replace the pseudorandom permutations π, π' with uniformly random permutations and add the corresponding terms to the security bound.

In their paper, Black and Rogaway prove that the distinguishing advantage against PMAC is at most $\sigma^2/2^n$, q being the number of message queries, ℓ being their maximal length, n being the block size, and σ the total sum of message blocks being queried. This was later improved by Minematsu and Matsushima [49] to $q^2\ell/2^n$, and then further to $q\sigma/2^n$ by Nandi and Mandal [52] (note that $q\sigma$ can be much less than $q^2\ell$, if the message lengths vary a lot). The most recent result on PMAC comes from Luykx *et al.* [42], who showed an attack with success probability roughly $\ell/2^n$. You can see all these results in relation to each other in Figure 5.1. In order to follow-up on Section 1.4, we mention that generic attacks on PMAC achieve success probability $\Omega(q^2/2^n)$, while $O(q^2\ell^2/2^n)$ could be considered a trivial upper bound.

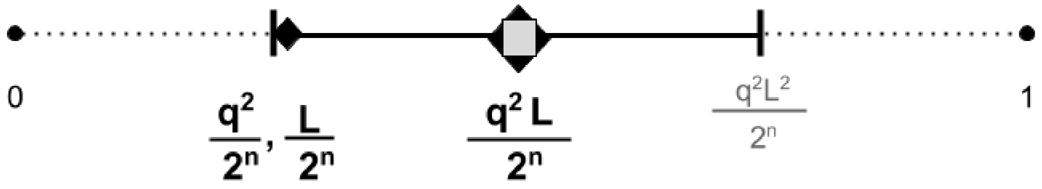


Figure 5.1: PMAC results overview.

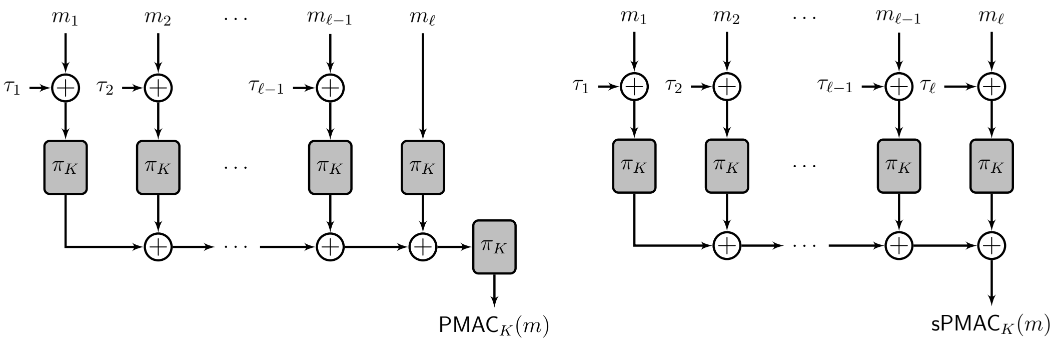


Figure 5.2: From PMAC to sPMAC.

In Section 5.2 we describe an attack on PMAC with advantage $O(q^2\ell/2^n)$, matching the bound of Nandi *et al.* [52], thus proving the exact security of PMAC. This attack is inspired by the one of Luykx *et al.* [42], but differs in the message construction. The former constructs 2 messages that are deterministically constructed based on the message length ℓ, leading to an attack with advantage $\ell/2^n$. This attack, however, cannot be extended to q queries. Our attack takes a different approach in message construction and can utilize the full power of q queries and therefore achieve the advantage of $q^2\ell/2^n$.

The existence of such attack shows that the security bounds of original PMAC cannot be further improved. Therefore, in Section 5.3, we look at different classes of masks $\tau_1, \ldots, \tau_\ell$ and whether they can boost the security to the desired $O(q^2/2^n)$ level. More concretely, we look at masks that are randomly distributed, 4-wise independent, and 2-wise independent (the original distribution is 1-wise independent). We prove that using either random masks, or 4-wise independent masks, we can indeed improve the security of PMAC to $q^2/2^n$, while we do not gain any security advantage by using 2-wise independent masks. The analysis of 3-wise independent masks is left as an open problem.

5.1 sPMAC

As mentioned at the beginning of this chapter, PMAC as we defined it is a somewhat simplified version of the actual original PMAC as proposed in [14]. We do not consider issues that deal with "imperfect" messages (such as padding messages that do not have full-block length) and, in general, issues that do not affect our security analysis. Additionally, we leave out the last message block, that is not permuted; again, this has no effect on the analysis. We also assume that the value L used to calculate the masks is sampled at random, even though technically is derived from the key. Same goes for the two permutations π and π' - we assume they are different, even though PMAC uses a single permutation. These changes simplify our analysis, while not having a significant effect on the outcome. For more details, see Chapter 7.

The last change we are going to describe is the removal of the outer permutation π'. Thanks to a result called the PRP/PRF switching lemma, we can replace π' with a PRF f, while adding a penalty to the security bound [12]. Then, we can use Lemma 1(i), which proves that distinguishing the output of f from random is as hard as provoking collisions on the input to f. Hence, we remove f alltogether and change our goal from distinguishing PMAC to finding collisions on the output of a new construction we call simplified PMAC, or just sPMAC (see Figure 5.2 for a comparison between the two constructions). Furthermore,

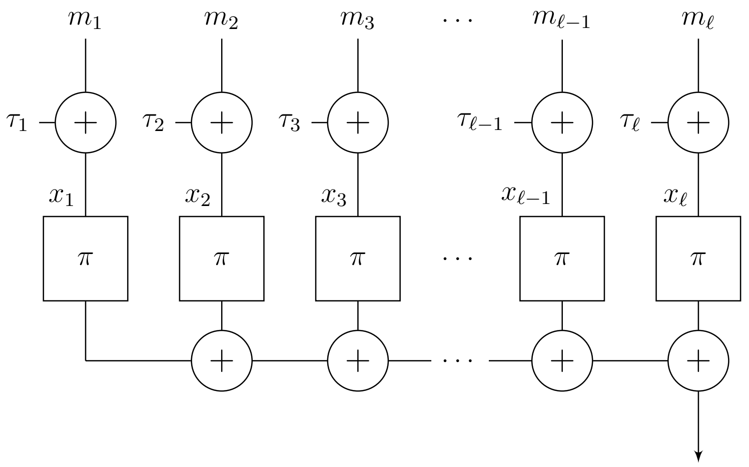


Figure 5.3: $\mathsf{sPMAC}_{\pi,\tau}(M)$.

by Lemma 1(ii), adaptivity does not help in provoking these collisions. Consequently, our analysis is static and we can submit the q attack queries together.

Formally, we define the simplified PMAC, $\mathsf{sPMAC}\colon \mathcal{P}_n \times \mathcal{F}_{\mathbb{N},n} \times \{0,1\}^{n*} \to \{0,1\}^n$ as

$$\mathsf{sPMAC}(\pi, \tau, m_1 \| \ldots \| m_\ell) := \bigoplus_{i=1}^{\ell} \pi(m_i \oplus \tau(i)) \ .$$

For a better understanding, see Figure 5.3. One can also show that PMAC as defined in Equation 5.1 can be derived from sPMAC by additionally encrypting the final output using an independent permutation π':

$$\mathsf{PMAC}(\pi, \pi', \tau, M) = \pi'(\mathsf{sPMAC}(\pi, \tau, M))$$

Sometimes, we write τ_i instead $\tau(i)$, and e.g., $\mathsf{sPMAC}_{\pi,\tau}(M)$ instead of $\mathsf{sPMAC}(\pi, \tau, M)$, or, if π, τ are clear from the context, simply $\mathsf{sPMAC}(M)$. Lastly, for an input message $M = m_1 \| \ldots \| m_\ell$, the following variables will be convenient to use later on

$$x_i := m_i \oplus \tau_i,\ \forall i \quad ; \quad \mathcal{X} := (x_1, \ldots, x_\ell) \tag{5.2}$$

These variables are also visually defined in the graph of sPMAC in Figure 5.3.

Because we are looking for collision, we need to work with pairs of messages $M = m_1 \| \ldots \| m_s, M' = m'_1 \| \ldots \| m'_{s'}$, and so $\mathcal{X}^*$ denotes the multiset

$$x_i := m_i \oplus \tau_i \ , \ x'_i := m'_i \oplus \tau_i,\ \forall i \qquad ; \qquad \mathcal{X}^* := (x_1, \ldots, x_s, x'_1, \ldots, x'_{s'}) \tag{5.3}$$

Then, a *cross-cancellation* for two messages M, M' (denoted $\mathsf{crCan}(M, M')$) occurs, if for their corresponding $\mathcal{X}^{*\downarrow}$, we have $\mathcal{X}^{*\downarrow} = \emptyset$ ($^\downarrow$ denotes the reduced set, see Chapter 2). It follows that a $\mathsf{crCan}(M, M')$ implies a collision on sPMAC, and hence on PMAC.

Lastly, for a given n, ℓ, and a distribution $\mathtt{T}_n$, we define the following quantity:

$$\theta(\ell, n, \mathtt{T}_n) = \max_{\substack{M \neq M' \\ |M|_n, |M'|_n \leq \ell}} \Pr_{\tau \leftarrow \mathtt{T}_n} \left[\{x_1, x_2, \ldots, x_{|M|_n}, x'_1, x'_2, \ldots, x'_{|M'|_n}\}^\downarrow = \emptyset \right] \ . \tag{5.4}$$

$\theta(\ell, n, \mathtt{T}_n)$ bounds the maximum probability over all pairs of distinct messages M, M' of maximum length ℓ that their reduced set $\mathcal{X}^{*\downarrow}$ is empty, and hence a cross-cancellation occurs. It is also the quantity we want to bound, as we show in the following lemma, relating it directly to sPMAC-collisions.

Lemma 5. *For any $n, \mathtt{T}_n$, and $\ell \leq 2^{n-2}$*

$$\mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(q,\ell) \leq \theta(\ell, n, \mathtt{T}_n) \cdot q^2 + \frac{q^2}{2^{n-1}} .$$

We do not give the proof here, but it can be found in Section 7.3. Intuitevely, $\theta(\ell, n, \mathtt{T}_n)$ is defined for a pair of messages, therefore, we take a union bound over all possible pairs out of q messages. This is the reason why we see the q^2 factor in the security bound. The $\frac{q^2}{2^{n-1}}$ bounds the probability that there is a collision of two messages as a result of xor-ing the permuted values x_i (which is very unlikely), taking the union bound again.

5.2 New attack on PMAC

In this section we describe the attack on sPMAC (and hence on PMAC) with success probability roughly $\ell q^2/2^n$.

Our attack can be used against sPMAC using a certain class of 1-wise independent mask distributions. Namely, we assume that the masks are derived as $\tau_i := p_i \cdot R$ for some progression $P = (p_1, \ldots, p_{2^n})$, where every $p_i \in \{0,1\}^n$ is distinct, and a value $R \xleftarrow{\$} \{0,1\}^n$, which we model as sampled uniformly at random. The most prominent progressions that satisfy the definition above are called Gray codes, and are in fact used in the original PMAC construction. A Gray code is an ordering $\gamma^\ell = \gamma^\ell_0 \gamma^\ell_1 \ldots \gamma^\ell_{2^\ell - 1}$ of $\{0,1\}^\ell$, for any $\ell \geq 1$, such that successive points differ in precisely one bit. The canonical Gray code from [14] is defined as follows:

$$\begin{aligned}
\gamma^1 &= (\gamma^1_0, \gamma^1_1) := (0,1)\\
\gamma^2 &= (\gamma^2_0, \gamma^2_1, \gamma^2_2, \gamma^2_3) := (00, 01, 11, 10)\\
&\vdots\\
\gamma^{\ell+1} &= (0\gamma^\ell_0, 0\gamma^\ell_1, \cdots, 0\gamma^\ell_{2^\ell-2}, 0\gamma^\ell_{2^\ell-1}, 1\gamma^\ell_{2^\ell-1}, 1\gamma^\ell_{2^\ell-2}, \cdots, 1\gamma^\ell_1, 1\gamma^\ell_0)
\end{aligned}$$

An important feature of this Gray code is that it forms an additive group in the field $GF(2^n)$ (where addition is the same as xor).

Description

In oder to better describe the attack, we introduce the following notation: given messages (i.e., attack queries) $M_1, \ldots, M_q$ of length ℓ blocks each, we denote the i-th block of the a-th message by $m^{(a)}_i$. We analogously define $x^{(a)}_i := m^{(a)}_i \cdot \tau_i = m^{(a)}_i \oplus p_i \cdot R$.

We describe a version of the attack that works against sPMAC with Gray codes. The more general version that works for any progression P as we defined it above can be found in Section 7.7. The main idea is that we look at messages that are constructed as

$$\begin{aligned}
M_1 &= (m_1, m_2, \ldots, m_\ell) = m^1 || m^1 || \ldots || m^1\\
&\ldots\\
M_q &= (m_1, m_2, \ldots, m_\ell) = m^q || m^q || \ldots || m^q
\end{aligned}$$

for some message blocks $m^1, \dots, m^q$. Then, we calculate the sets X_i (Equation 5.2) for each message M_i.

As a first step, we take M_1, M_2, for now denote them M, M', respectively. We want to analyse under what conditions the corresponding $\mathcal{X}^* = \emptyset$ (see Equation 5.3), and hence when they cause a crCan. Therefore, we first look at when x_1 collide with some block of the other message, x'_i. This happens precisely when

$$m_1 \oplus \tau_1 = m'_i \oplus \tau_i \tag{5.5}$$

$$\tau_1 \oplus \tau_i = m_1 \oplus m'_i \tag{5.6}$$

$$\gamma_1 \cdot R \oplus \gamma_i \cdot R = m_1 \oplus m'_i \tag{5.7}$$

$$R = \frac{m_1 \oplus m'_i}{\gamma_1 \oplus \gamma_i} \, . \tag{5.8}$$

This means there exists precisely one value R which causes $x_1 = x'_i$. Additionally, if we rearrange the elements in Equation 5.5, we can easily see that $x_i = x'_1$ (the messages are composed of equal message blocks). Now, we make use of the fact that $\gamma_1, \dots, \gamma_\ell$ form a group, and γ_1 is the identity element. It follows from Lagrange's Theorem that we can partition the set $\gamma_1, \dots, \gamma_\ell$ into pairs of elements γ_a, γ_b, such that their sum (xor) is equal to $\gamma_i = \gamma_i \oplus id = \gamma_i \oplus \gamma_1$. If we plug these pairs into Equation 5.8, we can conclude that $x_a = x'_b, x_b = x'_a$ for every pair of indices a, b as defined above. Because these cover the whole set of γ's, we can conclude that $\mathcal{X}^* = \emptyset$ for this particular value of R. Moreover, we have chosen the index i of γ_i at random from a set of size $\ell - 1$, hence we have $\ell - 1$ values of R that cause a collision.

The next question is, whether we can extend this attack to q messages. It turns out we can, if we choose $m^1, \dots, m^q$ carefully. Once again, we will not give the exact details and analysis, just the intuition. We use rejection-sampling method to find message blocks that maximize the number of R's that cause a collision among $M_1, \dots, M_q$. We also show that our algorithm finds such messages blocks in finite number of steps and is therefore sensible.

However, there is a slight problem with out attack. The original PMAC from [14] uses only

$$\gamma = (\gamma_1, \dots, \gamma_{2^n - 1}) = (\gamma_1^n, \gamma_2^n, \dots, \gamma_{2^n-1}^n) \, , \tag{5.9}$$

meaning they do not use the first element of the progression, $\gamma_0^n = 0$, for the mask construction. This however means that the set of masks does not form a group, as γ_0^n would be the identity element. We can fix this problem by slightly changing our approach and gaining a factor 2^{-1} in the security bound. A bit surprisingly, it turns out that one does not need $\gamma_1, \dots, \gamma_\ell$ to contain a full group. It is sufficient, if it contains a large enough coset of some subgroup of a group contained in $GF(2^n)$. The Gray code from [14] contains a coset of size $\ell/2$, which means that we can force a collision on sPMAC with probability roughly $\frac{q^2 \ell}{2^{n+1}}$.

We conclude the attack description with Algorithm 1, where we present the pseudocode for the attack on PMAC as defined in [14]. The adversary $\mathsf{A} := \mathsf{A}_{\ell,q,n}^{\mathcal{O}(\cdot)}$ is parametrized by variables ℓ, q, n (maximal length of messages, number of messages, size of message blocks), and expects to interact with an oracle $\mathcal{O}(\cdot)$ that is either PMAC, or a random function. If it sees a collision, it guesses the oracle to be PMAC, random function otherwise. As we said before, it succeeds with probability $\Omega(\frac{q^2 \ell}{2^{n+1}})$.

Algorithm 1: Attacker $\mathsf{A}^{\mathcal{O}(\cdot)}_{\ell,q,n}$ against PMAC

```
$e := \gamma_{\ell/2}$
$I'_S :=$ indices $(\ell/2+1)\dots\ell$
$\mathcal{U}_0 := \emptyset$
for $a := 1\dots q$ do
    repeat
        $\hat{m}^{(a)} \xleftarrow{\$} \{0,1\}^n$
    until $\left|\left\{\frac{\hat{m}^{(a)}\oplus\hat{m}^{(b)}}{e\oplus\gamma_i} : b \in [a-1], i \in I'_S\right\} \cap \mathcal{U}_{a-1}\right| \leq \frac{2(a-1)^3((\ell/2)-1)^2}{2^n}$
    $\mathcal{U}_a := \mathcal{U}_{a-1} \cup \left\{\frac{\hat{m}^{(a)}\oplus\hat{m}^{(b)}}{e\oplus\gamma_i} : b \in [a-1], i \in I'_S\right\}$
    $M_a := \emptyset$
for $a := 1\dots q$ do
    for $i := 1\dots\ell/2$ do
        $M_a := M_a||0^n$
    for $i := (\ell/2+1)\dots\ell$ do
        $M_a := M_a||\hat{m}^{(a)}$
for $i := 1\dots q$ do
    $\mathsf{Tag}_i := \mathcal{O}(M_i)$
for $i := 1\dots(q-1)$ do
    for $j := (i+1)\dots q$ do
        if $\mathsf{Tag}_i = \mathsf{Tag}_j$ then
            return 1
return 0
```

5.3 PMAC with k-wise independent masks

In this section we look at different distributions of masks $\tau_1,\dots,\tau_\ell$ with the goal of improving the security of PMAC.

2-wise independence In Section 5.2, we showed that the security of PMAC with the original distribution on masks from [14], which is only 1-wise independent, the security is just $\ell q^2/2^n$. As a first step, whether we can get a better security bound by switching to 2-wise independent distribution on masks.

For this reason, we slightly change the original distribution to make it 2-wise independent. The mask distribution we used in the attack can be rewritten as a member of the following family

$$\{i \to a \cdot p_i \mid a \in GF(2^n)\}\ ,$$

where p_i is the i-th Gray codeword and a stands for our value R. As before, $P = (p_1, p_2 \dots, p_{2^n})$ can be thought of any progression without repetitions. Now, let us modify this distribution by adding another field element to make it 2-wise independent:

$$\{i \to a \cdot p_i \oplus b \mid a, b \in GF(2^n)\}\ .$$

Note that this is the most standard 2-wise distribution. If you recall, in the previous section we were analyzing collisions, more precisely, we were looking for pairs of x_i, x'_j (defined in Equation 5.2), such that $x_i = x'_j$. Let us know look at one such pair computed using the 2-wise independent distribution we have just defined.

$$\begin{aligned} x_i &= x'_j \\ m_i \oplus a \cdot p_i \oplus b &= m'_j \oplus a \cdot p_j \oplus b \\ m_i \oplus a \cdot p_i &= m'_j \oplus a \cdot p_j \end{aligned}$$

As we see, the element b is automatically cancelled out, leaving us with exactly same equation as if we used the original 1-wise independent distribution. Therefore, we can conclude that using 2-wise independent mask distribution is not enough to improve the security of PMAC. More details can be found in Section 7.6.

Random masks Next, we look at a mask distribution from the other side of the spectrum and analyse what happens if the masks are chosen independently and uniformly at random.

It turns out that this distribution is enough to improve the security of PMAC to the $q^2/2^n$ level. Again, we do not give a formal proof which can be found in Section 7.4, but rather give an informal argument why this is true. For this thought experiment, let us assume 2 messages M, M'. Further assume that we have sampled all the masks $\tau_1, \ldots, \tau_{\ell-1}$ and consequently determined all the values in $\mathcal{X}^*$, but two corresponding to τ_ℓ. If there is to be a collision on the output of sPMAC for these two messages, then before τ_ℓ is sampled, the set $\mathcal{X}^{*^\downarrow}$ must contain precisely 2 elements, called them a, b. Intuitively, these are the two elements that are "waiting" to be matched with the last two elements determined from τ_ℓ. However, the value of a, b is fixed and τ_ℓ is chosen at random. Therefore, there are at most two values of τ_ℓ that match $m_\ell \oplus \tau_\ell$ to either a, or b. Therefore, the probability that M, M' collide is upper bounded by $2/2^n$. If we take the union bound over q messages, we conclude that the security bound for sPMAC (PMAC) with uniformly random masks against collisions is $\frac{2}{2^n}$.

4-wise independence The last mask-distribution we are going to look at is a 4-wise independent distribution. The argument will be somewhat similar to the one for random masks, also showing a bound roughly $\frac{q^2}{2^n}$.

Once more, we give only an intuition of the proof that is given in Section 7.5. The argument starts as before with the assumption that two messages M, M' collide with each other on the output of sPMAC with 4-wise independent masks. We then look at two pairings of values from $\mathcal{X}^*$ that are deterministically determined by choosing a specific pair of indices. The first index determines the block that is paired with x_1, call it a. Now, we define x_f to be the element with the lowest index, such that it does not share a mask with either a, or x_1. Then, the second index determines the index of a block that is paired with x_f. Clearly, each of the blocks is calculated based on a different mask. The probability they are assigned the value causing the pairings described above is at most 2^{-2n}. There are at most $4\ell^2$ different pairs of indices we can look at. If we combine these two bounds together, we see that when the mask distribution is 4-wise independent, a collision between two messages happens with probability $4/2^n$. All that remains is to take a union bound over q messages to reach the final security bound $O(q^2/2^n)$.

5.4 Further research

In this last section we would like to discuss some other works that are connected to PMAC, as well as interesting open problems.

Firstly, there are some newer variations of PMAC that show that by somewhat changing the construction, one can boost the security of PMAC [65, 66, 68] even beyond the $q^2/2^n$ birthday bound. These include PMAC+ [65], PMAC with parity [66], and PMACX [68]. These introduce major modifications to the original constructions, therefore we do not discuss them in more detail. Lastly, LightMAC [43] can be considered a PMAC-like construction.

There is also a later variant of PMAC due to Rogaway [58] called PMAC1, which for efficiency reasons deviates slightly from PMAC by using a different sequence for the γ_i values. It is not clear if our attack can be adapted to this case. Informally, we require the sequence of $\gamma_1, \ldots, \gamma_\ell$ to contain a large coset of a subgroup of $GF(2^n)$, and it's not clear if the sequence from [58] contains such a set. There is some experimental data that suggest that the masks of PMAC1 do not contain large cosets, meaning this variant is secure against our attack. Furthermore, as it does not contain large subgroups either, the attack from [42] does not apply as well. Hence, an interesting open question would be to prove a better security for PMAC1, or find a different attack.

As we mentioned before, we were unable to prove any result for PMAC using 3-wise independent masks. Consequently, analysis of this scenario constitutes another interesting open problem.

6 Paper 1

The Exact PRF-Security of NMAC and HMAC[1]

Peter Gaži, Krzysztof Pietrzak, Michal Rybár

IST Austria

August 2014

Abstract. NMAC is a mode of operation which turns a fixed input-length keyed hash function f into a variable input-length function. A practical single-key variant of NMAC called HMAC is a very popular and widely deployed message authentication code (MAC). Security proofs and attacks for NMAC can typically be lifted to HMAC.

NMAC was introduced by Bellare, Canetti and Krawczyk [Crypto'96], who proved it to be a secure pseudorandom function (PRF), and thus also a MAC, assuming that (1) f is a PRF and (2) the function we get when cascading f is weakly collision-resistant. Unfortunately, HMAC is typically instantiated with cryptographic hash functions like MD5 or SHA-1 for which (2) has been found to be wrong. To restore the provable guarantees for NMAC, Bellare [Crypto'06] showed its security based solely on the assumption that f is a PRF, albeit via a non-uniform reduction.

- Our first contribution is a simpler and *uniform* proof: If f is an ε-secure PRF (against q queries) and a δ-*non-adaptively* secure PRF (against q queries), then $\mathsf{NMAC}^{\mathsf{f}}$ is an $(\varepsilon + \ell q \delta)$-secure PRF against q queries of length at most ℓ blocks each.

- We then show that this $\varepsilon + \ell q \delta$ bound is basically tight. For the most interesting case where $\ell q \delta \geq \varepsilon$ we prove this by constructing an f for which an attack with advantage $\ell q \delta$ exists. This also violates the bound $O(\ell \varepsilon)$ on the PRF-security of NMAC recently claimed by Koblitz and Menezes.

- Finally, we analyze the PRF-security of a modification of NMAC called NI [An and Bellare, Crypto'99] that differs mainly by using a compression function with an additional keying input. This avoids the constant rekeying on multi-block messages in NMAC and allows for a security proof starting by the standard switch from a PRF to a random function, followed by an information-theoretic analysis. We carry out such an analysis, obtaining a tight $\ell q^2/2^c$ bound for this step, improving over the trivial bound of $\ell^2 q^2/2^c$. The proof borrows combinatorial techniques originally developed for proving the security of CBC-MAC [Bellare et al., Crypto'05]. We also analyze a variant of NI that does not include the message length in the last call to the compression function, proving a $\ell^{1+o(1)} q^2/2^c$ bound in this case.

Keywords: Message authentication codes, pseudorandom functions, NMAC, HMAC, NI.

[1] A preliminary version of this paper appears in the proceedings of CRYPTO 2014, this is the full version. This work was partly funded by the European Research Council under an ERC Starting Grant (259668-PSPC).

6.1 Introduction

NMAC is a mode of operation which transforms a keyed fixed input-length function $\mathsf{f} : \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ (with $b \geq c$) into a keyed variable input-length function $\mathsf{NMAC}^{\mathsf{f}} : \{0,1\}^{2c} \times \{0,1\}^{b*} \to \{0,1\}^c$ (where $\{0,1\}^{b*}$ denotes all bit strings whose length is a multiple of b) as

$$\mathsf{NMAC}^{\mathsf{f}}((K_1, K_2), M) := \mathsf{f}(K_2, \mathsf{Casc}^{\mathsf{f}}(K_1, M) \| 0^{b-c})$$

where $\mathsf{Casc}^{\mathsf{f}} : \{0,1\}^c \times \{0,1\}^{b*} \to \{0,1\}^c$ is the cascade (also known as Merkle-Damgård) construction

$$\mathsf{Casc}^{\mathsf{f}}(K_1, m_1 \| \dots \| m_\ell) := \mathsf{f}(\dots \mathsf{f}(\mathsf{f}(K_1, m_1), m_2) \dots m_\ell) \ .$$

HMAC is a variant of NMAC (we postpone its exact definition to Section 6.2.2) tweaked for applicability in practice. As security proofs for NMAC can typically be lifted to HMAC, it is usually sufficient to analyse the security of the cleaner NMAC construction, we will discuss this point further in Section 6.1.2.

NMAC and HMAC were introduced by Bellare, Canetti and Krawczyk in 1996 [7] and later standardized [39]. HMAC has also become very popular and widely used, being implemented in SSL, SSH, IPsec and TLS amongst other places. Although originally designed as a MAC, it is also often employed more broadly, as a pseudorandom function (PRF). This is the case for example when used for key-derivation in TLS and IKE (the Internet Key Exchange protocol of IPsec). This proliferation into practice motivates the need for a good understanding of the exact security guarantees provided by NMAC and HMAC when used as a PRF.

PRF-Security of NMAC. Bellare *et al.* [7] prove that NMAC is a secure PRF if (1) f is a PRF and (2) $\mathsf{Casc}^{\mathsf{f}}$ is weakly collision-resistant (WCR). This is a relaxed notion of collision resistance, where one requires that it is hard to find a pair of messages $M \neq M'$ such that $\mathsf{Casc}^{\mathsf{f}}(K, M) = \mathsf{Casc}^{\mathsf{f}}(K, M')$ under a random key K, given oracle access to $\mathsf{Casc}^{\mathsf{f}}(K, .)$ (but not K, as in the standard definition of collision resistance).

HMAC is typically instantiated with cryptographic hash functions like MD5 or SHA-1 playing the role of $\mathsf{Casc}^{\mathsf{f}}$. However, both of these have been found not to satisfy the WCR notion [61, 62], which renders the security proof from [7] irrelevant for this case. Despite that, no attacks (better than standard birthday attacks) are known for NMAC or HMAC when instantiated with MD5 or SHA-1 (though attacks on reduced round versions exist [37]).

Security without Collision-Resistance. To restore the provable security of NMAC, Bellare [5] investigates the security of NMAC dropping assumption (2), that is, assuming only that f is a secure PRF. The exact security statement from [5] is a bit technical, but it roughly states that if f is an ε-secure PRF (against an adversary running in time t and asking q queries) and a γ-secure PRF (against time $O(\ell)$ and 2 queries), then $\mathsf{NMAC}^{\mathsf{f}}$ is an $(\varepsilon + \ell q^2 \gamma)$-secure PRF against time t and q queries of length at most ℓ (in b-bit blocks). The security reduction is non-uniform, which means one has to be careful when deducing what this bound exactly means when instantiated in practice, we will discuss this further in Section 6.1.2.[2]

[2] We note that in a very recent update of the ePrint version of [5], Bellare observes that the proof in [5] can also give a uniform reduction, differing from the non-uniform case only in the running time of the 2-query adversary which then becomes t. The uniform bound given in this paper is better for most reasonable parameters.

6.1.1 Our Contributions

PRF-Security Proof for NMAC. Our first contribution is a simpler, uniform, and as we will show, basically tight proof for the PRF-security of $\mathsf{NMAC}^{\mathsf{f}}$ assuming only that f is a PRF: If f is an ε-secure PRF against q queries, then $\mathsf{NMAC}^{\mathsf{f}}$ is roughly $\ell q\varepsilon$-secure against q queries of length at most ℓ blocks each.

Our actual result is more fine-grained, and expresses the security in terms of both the adaptive and non-adaptive security of f. Let δ denote the PRF-security of f against q *non-adaptive* queries. Then our Theorem 3 states that $\mathsf{NMAC}^{\mathsf{f}}$ is roughly $(\varepsilon + \ell q\delta)$-secure (against q queries, each at most ℓ blocks). As non-adaptive adversaries are a subset of adaptive ones we have $\delta \leq \varepsilon$, and if $\delta \ll \varepsilon$, then our fine-grained bound is much better than the simpler $\ell q\varepsilon$ bound. The reduction works in the best running time one could hope for, its overhead being $\tilde{O}(\ell q)$.

The main technical part of our proof closely follows a proof by Bellare *et al.* [8] who show that if f is a secure fixed input-length PRF, then $\mathsf{Casc}^{\mathsf{f}}$ is a secure PRF if queried on prefix-free queries. We first observe that their proof also holds in the non-adaptive setting. Then we reduce the security of $\mathsf{NMAC}^{\mathsf{f}}$ against arbitrary adaptive queries to the security of $\mathsf{Casc}^{\mathsf{f}}$ against non-adaptive prefix-free queries.

Matching Attack for NMAC. In Section 6.3.2 we prove that the above lower bound is basically tight. From any PRF, we construct another PRF f for which $\mathsf{NMAC}^{\mathsf{f}}$ can be broken with advantage $\Theta(\ell q\delta)$. This shows that our bound is tight for the practically most important case when $\ell q\delta$ is larger (or at least comparable) to ε.

We also consider the case where $\varepsilon \gg \ell q\delta$, that is, when the PRF has much better security against non-adaptive than adaptive distinguishers. We observe that for any ε, we can use a result due to Pietrzak [55] who shows that cascading non-adaptively secure PRFs does not give an adaptively secure PRF in general, to construct an ε-secure f where $\mathsf{NMAC}^{\mathsf{f}}$ can be broken with advantage $\Theta(\varepsilon^2)$. This only shows the ε term is necessary if ε is constant as then $\Theta(\varepsilon) = \Theta(\varepsilon^2) = \Theta(1)$. We conjecture that $\Theta(\varepsilon^2)$ is the correct value, and the ε term in the lower bound can be improved to $\Theta(\varepsilon^2)$ using security amplification techniques along the lines of [48, 60].

PRF-Security Proof for NI. The main difficulty in security analyses of $\mathsf{NMAC}^{\mathsf{f}}$ and $\mathsf{HMAC}^{\mathsf{f}}$ based on the PRF-security of the underlying compression function f is that both these constructions are constantly rekeying f during the evaluation of $\mathsf{Casc}^{\mathsf{f}}$, using the output from the last invocation as the key for the next one. This prevents the proof approach typically applied to constructions that use a PRF f under a fixed random secret key, where the analysis starts by replacing the PRF with an ideal random function (introducing an error that is upper-bounded by the PRF-security of f) and proceeds by a fully information-theoretic argument.

To circumvent this issue, as our third contribution we investigate the PRF-security of the nested iterated (NI) construction introduced in [4]. The construction NI^{h} is very similar to $\mathsf{NMAC}^{\mathsf{f}}$, but is based on a compression function h that (compared to f) takes an additional k-bit input which is used for keying instead of the chaining input: NI^{h} uses h under the same key throughout the whole cascade. Additionally, it includes the length of the message in the input to the final, outer h-call. The modified keying allows for the simple switching argument from PRF to a random function. We focus on enhancing the information-theoretic analysis that follows this switch and prove an essentially tight $\ell q^2/2^c$ bound for this step, improving significantly over the trivial bound of $\ell^2 q^2/2^c$. For completeness, we also consider the modification of NI that does not include the message length in the last h-call and show a security bound of $\ell d'(\ell)q^2/2^c$ for this case, where $d'(\ell) \approx \ell^{1/\ln\ln\ell}$ denotes the maximum number of divisors of any positive integer not greater than ℓ. Our proofs employ combinatorial techniques originally developed for proving the security of CBC-MAC [11], considerably adapted for our setting.

6.1.2 More Related Work

Indifferentiability. In practice, the HMAC construction is sometimes used in a setting where stronger guarantees than PRF-security are needed. Motivated by this, recent work [22] investigates the indifferentiability [47, 17] of HMAC from a (keyed) random oracle. This result is incomparable to ours: While the stronger notion of indifferentiability covers the settings where HMAC is not used as a PRF, the bound achieved in [22] is understandably much weaker, being $\Theta(\ell^2q^2/2^c)$.

Generic Attacks. There is also a recent line of work investigating generic attacks against iterated hash-based MACs [53, 40, 50, 54]. These works present various attacks against MACs (e.g. related-key attack, universal forgeries, state recovery) that do not exploit the inner structure and potential weaknesses of the compression function, instead they rely solely on the iterative structure of the MACs.

Another look at [38]. As already mentioned, Bellare [5] proved that $\mathsf{NMAC}^{\mathsf{f}}$ is an $(\varepsilon + \ell q^2\gamma)$-secure PRF against q queries if f is ε-secure against q queries, and γ-secure against 2 queries. In a recent paper [38], Koblitz and Menezes present a criticism of the way [5] discusses the practical implications of this result. In a nutshell, Bellare estimates that for a well-designed PRF the γ term is roughly $t/2^c$ (for a 2-query adversary running in time t), but as this γ is derived in a non-uniform way, it is in the order of $2^{-c/2}$ already for constant t.

At the time when [5] appeared, the fact that non-uniform attacks can distinguish any pseudorandom object generated using a c-bit key with advantage $2^{-c/2}$ in constant time was not widely known in the crypto community[3] and overoptimistic estimates for the exact security implied by non-uniform reductions have appeared in numerous papers.[4] This changed at the latest with the Crypto 2010 paper [18], who discuss this issue in detail and attribute such generic non-uniform attacks to the 1992 paper by Alon *et al.* [2].

The paper [38] also claimed that HMAC is an $\varepsilon\ell$-secure PRF, a bound that is falsified by an attack given in this paper. In response, [38] was updated to take account of this by employing a non-standard definition of a PRF for the underlying compression function. We believe that the updated claim can be obtained via a simpler proof from [8].

HMAC vs NMAC. The proofs in this paper consider NMAC. There is a standard reduction of HMAC-to-NMAC PRF-security given by Bellare [5], albeit under some additional

[3] Let us stress that this only holds for pseudorandom objects which do not require additional *public* randomness, such as PRFs. This does not extend to weak PRFs, which are defined like PRFs but the adversary only sees the output on random inputs.

[4] This should not be confused with the (less trivial, but in the crypto community long well-known) fact that non-uniform generic attacks beating simple brute-force key search exist for “large” running times, as shown in a classical result by Hellman [30]. Hellman's result for example implies that there almost certainly exist key-recovery attacks against AES with a k bit key (k being 128, 192 or 256) which succeed with probability at least 1/2 and run in time $\approx 2^{2k/3}$, and in particular much less than 2^k required for brute-force key search.

requirements on the underlying compression function f. Informally, one needs to assume that f is a PRF even when keyed through the b-bit data input, as opposed to being keyed by the c-bit chaining variable. Moreover, security of the single-key version of HMAC requires the PRF to be secure under a specific class of related-key attacks. Formally, the reductions are given in Lemmas 5.1 and 5.2 in the full version of [5] for the case of double- and single-keyed HMAC, respectively. Since these reductions only relate to NMAC via its PRF-security, they apply to our result in a blackbox way, thus giving clear statements also for HMAC.

6.2 Preliminaries

Basic Definitions. We reserve the letter λ do denote the empty string. We use $\{0,1\}^{b*} := \bigcup_{z\geq 0}\{0,1\}^{bz}$ to denote the set of all bitstrings whose length is a multiple of b. $\mathcal{F}(b,c)$ (resp. $\mathcal{F}(b*,c)$) denotes the sets of all functions from $\{0,1\}^b$ to $\{0,1\}^c$ (resp. from $\{0,1\}^{b*}$ to $\{0,1\}^c$). We denote by $\mathsf{Pow}(\mathcal{S})$ the power set of the set $\mathcal{S}$. For an integer n, $d(n) = |\{i \in \mathbb{N} : i \mid n\}|$ is the number of its positive divisors and

$$d'(n) := \max_{n'\in\{1,\ldots,n\}} |\{d \in \mathbb{N} : d \mid n'\}| \approx n^{1/\ln\ln n}$$

is the maximum, over all positive integers $n' \leq n$, of the number of positive divisors of n'. More precisely, we have $\forall\varepsilon > 0\ \exists n_0\ \forall n > n_0 \colon d(n) < n^{(1+\varepsilon)/\ln\ln n}$ [29]. All logarithms considered in the paper are base 2 unless indicated otherwise.

Random Variables and Experiments. Random variables and concrete values they can take are usually denoted by upper-case letters $X, Y, \ldots$ and lower-case letters $x, y, \ldots$, respectively. If $\mathcal{M}$ is a distribution (respectively, a set), then we denote by $X \leftarrow \mathcal{M}$ sampling the random variable X according to $\mathcal{M}$ (respectively, choosing it uniformly at random from $\mathcal{M}$). For events A and B and random variables U and V with ranges $\mathcal{U}$ and $\mathcal{V}$, respectively, we denote by $\mathsf{Pr}_{UA|VB}$ the corresponding conditional probability distribution, seen as a (partial) function $\mathcal{U} \times \mathcal{V} \to [0,1]$. The value $\mathsf{Pr}_{UA|VB}(u,v) = \mathsf{Pr}[U = u \wedge A | V = v \wedge B]$ is well-defined for all $u \in \mathcal{U}$ and $v \in \mathcal{V}$ such that $\mathsf{Pr}_{VB}(v) > 0$ and undefined otherwise. Two probability distributions Pr_U and $\mathsf{Pr}_{U'}$ on the same set $\mathcal{U}$ are equal, denoted $\mathsf{Pr}_U = \mathsf{Pr}_{U'}$, if $\mathsf{Pr}_U(u) = \mathsf{Pr}_{U'}(u)$ for all $u \in \mathcal{U}$. Conditional probability distributions are equal if the equality holds for all arguments for which both of them are defined. To emphasize the random experiment $\mathcal{E}$ in consideration, we sometimes write it in the superscript, e.g. $\mathsf{Pr}^{\mathcal{E}}_{U|V}(u,v)$. If the distribution of a random variable U is clear from the context, we also sometimes write Pr^U to refer to the random experiment where U is chosen according to its distribution.

6.2.1 Random Systems

To present our results we make use of Maurer's random systems framework [46], which we now introduce in a self-contained exposition sufficient to follow the rest of the paper. This choice is a matter of authors' taste, we believe that the results could also be obtained using the game-playing framework [12].

We start by observing that the input-output behavior of any kind of reactive discrete system with inputs in $\mathcal{X}$ and outputs in $\mathcal{Y}$ can be described by an infinite family of functions specifying, for each $i \geq 1$, the probability distribution of the system's i-th output $Y_i \in \mathcal{Y}$, given the values of the first i inputs $X^i \in \mathcal{X}^i$ and the previous $i-1$ outputs $Y^{i-1} \in \mathcal{Y}^{i-1}$. Using this viewpoint, we say that an $(\mathcal{X}, \mathcal{Y})$-*(random) system* $\mathbf{F}$ is an infinite sequence of functions $\mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}} \colon \mathcal{Y} \times \mathcal{X}^i \times \mathcal{Y}^{i-1} \to [0,1]$ such that $\sum_{y_i} \mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}(y_i, x^i, y^{i-1}) = 1$ for all $i \geq 1$, $x^i \in \mathcal{X}^i$ and $y^{i-1} \in \mathcal{Y}^{i-1}$. Note that $\mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}$ by itself does not represent a (conditional) probability distribution in any particular random experiment with well-defined random variables Y_i, X^i, Y^{i-1} until the system is connected to a distinguisher (see below), in which case these random variables will exist and take the role of the transcript. We shall typically define discrete systems by a high level description, as long as the resulting conditional probability distributions could be derived easily from this description. Two systems $\mathbf{F}$ and $\mathbf{G}$ are called *equivalent* (denoted $\mathbf{F} \equiv \mathbf{G}$) if their input-output behaviors are the same, i.e., $\mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}} = \mathsf{p}^{\mathbf{G}}_{Y_i|X^iY^{i-1}}$ for all $i \geq 1$.

A system $\mathbf{F}$ might often be used as a component (subsystem) in a construction $\mathbf{C}^{(\cdot)}$, resulting in the composed system $\mathbf{C}^{\mathbf{F}}$. $\mathbf{F} \triangleright \mathbf{G}$ denotes the serial composition of systems: every input to $\mathbf{F} \triangleright \mathbf{G}$ is fed to $\mathbf{F}$, its output is fed to $\mathbf{G}$ and the output of $\mathbf{G}$ is used as the output of $\mathbf{F} \triangleright \mathbf{G}$. In case $\mathbf{G}$ takes as inputs longer bitstrings than $\mathbf{F}$ outputs (as will be the case in the definition of NMAC), the construction $\mathbf{F} \triangleright \mathbf{G}$ pads the outputs of $\mathbf{F}$ with trailing zeroes before passing them to $\mathbf{G}$.

Examples. We denote by $\mathbf{R}$ a system that provides access to a function chosen uniformly at random from the set of all functions with domain $\{0,1\}^{b*}$ and range $\{0,1\}^c$. (This unusual domain slightly deviates from the standard definition of $\mathbf{R}$ in the random-systems literature, but will be advantageous for our exposition.) Similarly, for a finite domain $\{0,1\}^b$ we denote by $\mathbf{r}$ a system realizing a function chosen uniformly from $\mathcal{F}(b,c)$. Finally, we also consider a system $\mathbf{f}$ realizing a function chosen uniformly from $\mathcal{F}(c+b,c)$. We refer to $\mathbf{R}$, $\mathbf{r}$ and $\mathbf{f}$ as a uniformly random function (URF), a fixed input-length URF, and an ideal compression function, respectively. In each case the parameters b and c will be clear from the context.

Distinguishers and Adversaries. A *distinguisher* $\mathbf{D}$ for an $(\mathcal{X}, \mathcal{Y})$-random system asking q queries is a $(\mathcal{Y}, \mathcal{X})$-random system which is "one query ahead:" its input-output behavior is defined by the conditional probability distributions of its queries $\mathsf{p}^{\mathbf{D}}_{X_i|X^{i-1}Y^{i-1}}$ for all $1 \leq i \leq q$. (Its first query is determined by $\mathsf{p}^{\mathbf{D}}_{X_1}$.) After the distinguisher asks all q queries, it outputs a bit W_q depending on the transcript (X^q, Y^q). Given a random system $\mathbf{F}$ and a distinguisher $\mathbf{D}$, we denote by $\mathbf{DF}$ the random experiment where $\mathbf{D}$ interacts with $\mathbf{F}$, with the distributions of the transcript (X^q, Y^q) and of the bit W_q being uniquely defined by their conditional probability distributions. For two $(\mathcal{X}, \mathcal{Y})$-random systems $\mathbf{F}$ and $\mathbf{G}$, the *distinguishing advantage* of $\mathbf{D}$ in distinguishing systems $\mathbf{F}$ and $\mathbf{G}$ by q queries is the quantity $\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{G}) = |\mathsf{Pr}^{\mathbf{DF}}_{W_q}(1) - \mathsf{Pr}^{\mathbf{DG}}_{W_q}(1)|$ and the maximal distinguishing advantage over all distinguishers asking q queries is denoted by $\Delta_q(\mathbf{F}, \mathbf{G}) = \max_{\mathbf{D}} \Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{G})$ (with $\mathbf{D}$ ranging over all such distinguishers).

As opposed to the information-theoretic notion of a distinguisher, we often need to consider an attacker with restricted computational resources. Although such an attacker also participates in a distinguishing experiment, to emphasize this restriction we call it an *adversary* and denote using a sans-serif symbol (e.g. A). Note that a computationally restricted adversary implicitly defines a random system by its input-output behavior and hence any notation defined for information-theoretic distinguishers is also well-defined for such an adversary. We often restrict the computational power of an adversary by its running time, for this we assume some reasonable fixed model of computation.

Monotone Conditions. For a random system $\mathbf{F}$, we often consider an internal *monotone condition* defined on it. Such a condition is initially satisfied (true), but once it gets violated, it cannot become true again (hence the name monotone). We use such conditions to capture whether the behavior of the system meets some additional requirement (e.g. distinct outputs, consistent outputs) or this was already violated during the interaction that occurred so far. A monotone condition is formalized by a sequence of events $\mathcal{A} = A_0, A_1, \ldots$ such that A_0 always holds, and A_i holds if the condition holds after answering the i-th query. The probability that a distinguisher $\mathbf{D}$ issuing q queries to $\mathbf{F}$ makes a monotone condition $\mathcal{A}$ fail in the random experiment $\mathbf{DF}$ is denoted by $\nu^{\mathbf{D}}(\mathbf{F}, \overline{A_q}) = \mathsf{Pr}^{\mathbf{DF}}(\overline{A_q})$ and maximum over all such distinguishers is denoted by $\nu(\mathbf{F}, \overline{A_q}) = \max_{\mathbf{D}} \nu^{\mathbf{D}}(\mathbf{F}, \overline{A_q})$. We also define $\mu(\mathbf{F}, \overline{A_q}) = \max_{x^q} \mathsf{Pr}^{\mathbf{F}}_{\overline{A_q}|X^q}(x^q)$ to be the maximal probability of violating the condition $\mathcal{A}$ by a sequence of q non-adaptive queries.

For a random system $\mathbf{F}$ with a monotone condition $\mathcal{A} = A_0, A_1, \ldots$ and a random system $\mathbf{G}$, we say that $\mathbf{F}$ *conditioned on* $\mathcal{A}$ *is equivalent to* $\mathbf{G}$, denoted $\mathbf{F}|\mathcal{A} \equiv \mathbf{G}$, if $\mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}A_i} = \mathsf{p}^{\mathbf{G}}_{Y_i|X^iY^{i-1}}$ for $i \geq 1$, for all arguments for which $\mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}A_i}$ is defined. Intuitively, this captures the fact that as long as the condition $\mathcal{A}$ holds in $\mathbf{F}$, it behaves the same as $\mathbf{G}$. The following useful claims were given in [46], see also [32] for the proof of claim (ii) and [45] for further discussion.

Lemma 6. *Let $\mathbf{F}$ and $\mathbf{G}$ be random systems, let $\mathcal{A}$ be a monotone condition defined on $\mathbf{F}$, let $\mathbf{D}$ be a distinguisher asking q queries. Then:*

(i) [46, Lemma 7] If $\mathbf{F}|\mathcal{A} \equiv \mathbf{G}$ then $\Delta^{\mathbf{D}}(\mathbf{F}, \mathbf{G}) \leq \nu^{\mathbf{D}}(\mathbf{F}, \overline{A_q})$.

(ii) [46, Theorem 2] If $\mathsf{p}^{\mathbf{F}}_{A_i|X^iY^{i-1}A_{i-1}} = \mathsf{p}^{\mathbf{F}}_{A_i|X^iA_{i-1}}$ for all $i \geq 1$, then $\nu(\mathbf{F}, \overline{A_q}) = \mu(\mathbf{F}, \overline{A_q})$.

6.2.2 Message Authentication Codes and PRFs

The standard security requirement for a MAC is *unforgeability under chosen-message attack*. However, it is well-known that any PRF attains this property [9], hence in this paper we focus on PRF-security of the analyzed constructions.

If the first component of the input to a function f is to be seen as a key, we sometimes call f a *keyed* function to emphasize this. For a keyed function $f\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ under a key $k \in \mathcal{K}$ we often write $f_k(\cdot)$ instead of $f(k, \cdot)$. A variable input-length keyed function $\mathsf{G} : \{0,1\}^c \times \{0,1\}^{b*} \to \{0,1\}^c$ is an:

- $(\varepsilon, t, q, \ell)$*-secure PRF*, if for any adversary A running in time t and making at most q queries, each of length at most ℓ (in b-bit blocks), a URF $\mathbf{R}\colon \{0,1\}^{b*} \to \{0,1\}^c$ and a uniformly random key $K \leftarrow \{0,1\}^c$, we have $\Delta^{\mathsf{A}}(\mathsf{G}_K, \mathbf{R}) \leq \varepsilon$.
- $(\varepsilon, t, q, \ell)$*-NA-secure PRF*, if the above is true for all adversaries A that choose their queries non-adaptively (i.e., A has to choose its q queries before seeing any of the outputs).
- $(\varepsilon, t, q, \ell)$*-PF-secure PRF*, if the above is true for all adversaries A that choose their queries to be prefix-free (i.e., no query is a prefix of another query).
- $(\varepsilon, t, q, \ell)$*-NA-PF-secure PRF*, if the above is true for all adversaries A that choose queries *both* non-adaptively and prefix-free.

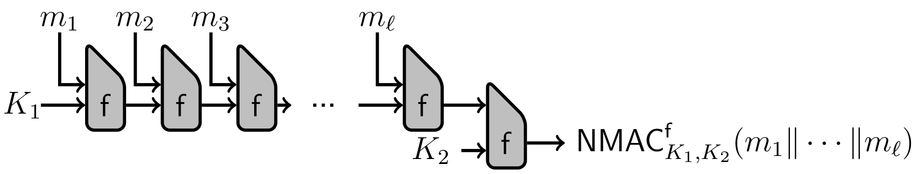


Figure 6.1: The construction $\mathsf{NMAC}^{\mathsf{f}}_{K_1,K_2}$.

For fixed input-length functions, we define analogous notions by omitting the parameter ℓ and distinguishing from $\mathbf{r}$ instead of $\mathbf{R}$. Moreover, we refer to an adversary A as an $(\varepsilon, t, q, \ell)$-PRF adversary against G if it runs in time t, asks at most q queries each consisting of at most ℓ blocks, and achieves the advantage $\Delta^{\mathsf{A}}(\mathsf{G}_K, \mathbf{R}) = \varepsilon$. We refer analogously to adversaries for the other PRF-notions defined above.

For a keyed function $\mathsf{f} : \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ we denote with $\mathsf{Casc}^{\mathsf{f}} : \{0,1\}^c \times \{0,1\}^{b*} \to \{0,1\}^c$ the cascade construction (also known as Merkle-Damgård) built from f as

$$\mathsf{Casc}^{\mathsf{f}}(K, m_1\|\ldots\|m_\ell) := y_\ell \quad \text{where} \quad y_0 := K \quad \text{and for} \quad i \geq 1 \;:\; y_i := \mathsf{f}(y_{i-1}, m_i)\;,$$

in particular $\mathsf{Casc}^{\mathsf{f}}(K, \lambda) := K$.

The construction $\mathsf{NMAC}^{\mathsf{f}} \colon (\{0,1\}^c)^2 \times \{0,1\}^{b*} \to \{0,1\}^c$ is derived from $\mathsf{Casc}^{\mathsf{f}}$ by adding an additional, independently keyed application of f at the end. It assumes that the domain sizes of f satisfy $b \geq c$ and the output of the cascade is padded with zeroes before the last f-call. Formally,

$$\mathsf{NMAC}^{\mathsf{f}}((K_1, K_2), M) := \mathsf{f}(K_2, \mathsf{Casc}^{\mathsf{f}}(K_1, M)\|0^{b-c})$$

or $\mathsf{NMAC}^{\mathsf{f}}_{K_1,K_2} := \mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathsf{f}_{K_2}$, see Figure 6.1. Note that practical MD-based hash functions take as input arbitrary-length bitstrings and then pad them to a multiple of the block length, often including the message length in the so-called MD-strengthening. This padding then also appears in NMAC (and HMAC) but since it does not affect any of our arguments, we take the customary shortcut and our definition of NMAC above (resp. HMAC below) actually corresponds to the generalized constructions denoted as GNMAC (resp. GHMAC) in [5] where this step is also justified in detail.

$\mathsf{HMAC}^{\mathsf{f}}$ is a practice-oriented version of $\mathsf{NMAC}^{\mathsf{f}}$, where the two keys (K_1, K_2) are derived from a single key $K \in \{0,1\}^b$ by xor-ing it with two fixed b-bit strings ipad and opad. In addition, the keys are not given through the key-input of the compression function f, but are prepended to the message instead. This allows for the usage of existing implementations of hash functions that contain a hard-coded initialization vector IV. Formally:

$$\begin{aligned} \mathsf{HMAC}^{\mathsf{f}}(K, m) \;&:=\; \mathsf{Casc}^{\mathsf{f}}(\mathsf{IV}, K_2\|\mathsf{Casc}^{\mathsf{f}}(\mathsf{IV}, K_1\|m)\|\mathsf{fpad}) \\ &\;\; \text{where } (K_1, K_2) := (K \oplus \mathsf{ipad}, K \oplus \mathsf{opad}) \end{aligned}$$

and fpad is a fixed $(b-c)$-bit padding not affecting the security analysis. (Technically, [39] allows for arbitrary length of the key K: a key shorter than b bits is padded with zeroes before applying the xor transformations, a longer key is first hashed.) As discussed in Section 6.1.2, we can focus on the PRF-security of NMAC as it translates to analogous results for HMAC under the assumptions stated in [5].

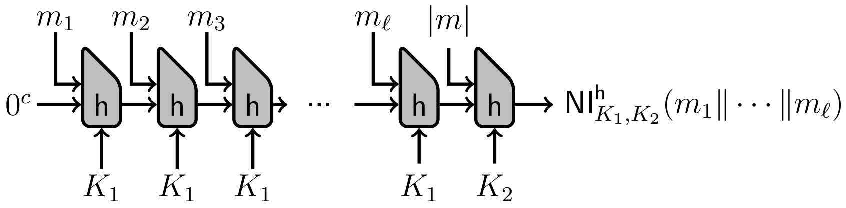


Figure 6.2: The construction $\mathsf{NI}^{\mathsf{h}}_{K_1,K_2}$.

Finally, we also introduce the nested iterated (NI) construction defined in [4]. For this, we consider a keyed compression function $\mathsf{h}\colon \{0,1\}^k \times \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$. When such h is used in a cascading construction, its c-bit and b-bit inputs are used for the chaining value and the next block, respectively. In contrast to the function f considered above, h has an additional k-bit input that is used for keying. Formally, for such h we define the *nested iterated* construction $\mathsf{NI}^{\mathsf{h}}\colon (\{0,1\}^k)^2 \times \{0,1\}^{b*} \to \{0,1\}^c$ as

$$\mathsf{NI}^{\mathsf{h}}_{K_1,K_2}(m) := \mathsf{h}_{K_2}(\mathsf{Casc}^{\mathsf{h}_{K_1}}_{\mathbf{0}}(m), |m|)$$

where $\mathbf{0}$ denotes the all zero bitstring 0^c and $|m|$ is the length of m encoded as a b-bit string. Alternatively, for a function $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ and a key K we will denote by $\mathsf{LenCasc}^{\mathsf{f}}_K$ a system that given a message m outputs the pair $(\mathsf{Casc}^{\mathsf{f}}_K(m), |m|)$. This allows us to describe NI equivalently as $\mathsf{NI}^{\mathsf{h}}_{K_1,K_2} := \mathsf{LenCasc}^{\mathsf{h}_{K_1}}_{\mathbf{0}} \triangleright \mathsf{h}_{K_2}$, see also Figure 6.2. For a detailed discussion of the relationship of NI to NMAC, see [4]. For completeness, we also consider the modified version of NI that replaces the message length $|m|$ in the last (outer) call of the compression function by the constant bitstring 0^b, we denote this variant as $\mathsf{NI2}$. Formally, we have

$$\mathsf{NI2}^{\mathsf{h}}_{K_1,K_2}(m) := \mathsf{h}_{K_2}(\mathsf{Casc}^{\mathsf{h}_{K_1}}_{\mathbf{0}}(m), 0^b)$$

or $\mathsf{NI2}^{\mathsf{h}}_{K_1,K_2} := \mathsf{ZCasc}^{\mathsf{h}_{K_1}}_{\mathbf{0}} \triangleright \mathsf{h}_{K_2}$, where $\mathsf{ZCasc}^{\mathsf{f}}_K$ a system that given a message m outputs the pair $(\mathsf{Casc}^{\mathsf{f}}_K(m), 0^b)$.

6.3 PRF-Security of NMAC

In this section we analyze the PRF security of $\mathsf{NMAC}^{\mathsf{f}}$ in terms of the PRF-security of the underlying function f.

6.3.1 Security Lower Bound

Before moving to the $\mathsf{NMAC}^{\mathsf{f}}$ construction, we start by stating a lower bound on the security of the cascade $\mathsf{Casc}^{\mathsf{f}}$ when queried on prefix-free inputs. A similar statement has already been proven in [8], and we follow their proof, modifying it where necessary to obtain security against *non-adaptive* adversaries, assuming only *non-adaptive security* of the underlying compression function f. The proof of Proposition 3 is given in Appendix 6.5.1.

Proposition 3 ($\mathsf{Casc}^{\mathsf{f}}$ as a NA-PF-PRF)**.** *Let $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ be a compression function. There exists an explicit reduction T (described in the proof) such that* *for any $(\varepsilon', t', q, \ell)$-NA-PF-PRF adversary* A *against* $\mathsf{Casc}^{\mathsf{f}}$, T^{A} *is an* $(\varepsilon_{\mathsf{na}}, t, q)$*-NA-PRF adversary against* f *such that*

$$\varepsilon' \le \ell q \varepsilon_{\mathsf{na}} \qquad \text{and} \qquad t = t' + \tilde{O}(\ell q) \ .$$

This allows us to present our main result in this section, which relates the adaptive PRF-security of the construction $\mathsf{NMAC}^{\mathsf{f}}$ to both the adaptive and non-adaptive PRF-security of f.

Theorem 3 ($\mathsf{NMAC}^{\mathsf{f}}$ as a PRF)**.** *If* $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ *is an* (ε, t, q)*-secure PRF and an* $(\varepsilon_{\mathsf{na}}, t, q)$*-NA-secure PRF, then* $\mathsf{NMAC}^{\mathsf{f}}$ *is an* $(\varepsilon', t', q, \ell)$*-secure PRF with*

$$\varepsilon' = \varepsilon + (\ell+1) q \varepsilon_{\mathsf{na}} + \frac{q^2}{2^c} \qquad \text{and} \qquad t = t' + \tilde{O}(\ell q) \ . \tag{6.1}$$

The reduction is uniform. Concretely, there exist explicit reductions T_1 *and* T_2 *(described in the proof) such that for any* $(\varepsilon', t', q, \ell)$*-PRF adversary* A *against* $\mathsf{NMAC}^{\mathsf{f}}$*,*

1. $\mathsf{T}_1^{\mathsf{A}}$ *is an* (ε, t, q)*-PRF adversary against* f*,*
2. $\mathsf{T}_2^{\mathsf{A}}$ *is an* $(\varepsilon_{\mathsf{na}}, t, q)$*-NA-PRF adversary against* f*,*

and their parameters satisfy equations (6.1).

Proof. Let A be a PRF-adversary running in time t' and asking q queries, each of length at most ℓ blocks. Let $\mathbf{r}\colon \{0,1\}^b \to \{0,1\}^c$, $\mathbf{R}\colon \{0,1\}^{b*} \to \{0,1\}^c$ and $K = (K_1, K_2) \leftarrow \{0,1\}^c \times \{0,1\}^c$ denote a fixed input-length URF, a URF and a key pair chosen independently at random, respectively.

We turn A into an adversary $\mathsf{T}_1^{\mathsf{A}}$ against the PRF-security of f_K as follows: Given access to g (which is either f_K or $\mathbf{r}$), sample some key K_1 at random, and then invoke A, answering its queries with $\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright g$. Finally, output the decision bit of A. Clearly we have $\Delta^{\mathsf{A}}(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathsf{f}_{K_2}, \mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}) = \Delta^{\mathsf{T}_1^{\mathsf{A}}}(\mathsf{f}_K, \mathbf{r})$ and if we denote $\Delta^{\mathsf{T}_1^{\mathsf{A}}}(\mathsf{f}_K, \mathbf{r})$ by ε then using triangle inequality we get

$$\Delta^{\mathsf{A}}(\mathsf{NMAC}^{\mathsf{f}}_K, \mathbf{R}) = \Delta^{\mathsf{A}}(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathsf{f}_{K_2}, \mathbf{R}) \le \varepsilon + \Delta^{\mathsf{A}}(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}, \mathbf{R}) \ .$$

In the experiment where A interacts with $\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}$, let C_i denote the event that during the first i queries to $\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}$, for any two distinct queries M and M' the values $\mathsf{Casc}^{\mathsf{f}}_{K_1}(M)$ and $\mathsf{Casc}^{\mathsf{f}}_{K_1}(M')$ (inputs to the final $\mathbf{r}$-call) are also distinct. As long as the monotone condition $\mathcal{C} = C_0, C_1, \ldots$ remains satisfied, the responses of $\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}$ to distinct queries are equivalent to outputs of $\mathbf{r}$ on distinct inputs, and thus independent, uniformly random values, in particular $(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r})|\mathcal{C} \equiv \mathbf{R}$. We can therefore apply Lemma 6(i) to conclude that distinguishing $\mathsf{Casc}^{\mathsf{f}} \triangleright \mathbf{r}$ from a URF $\mathbf{R}$ is at least as hard as making the condition $\mathcal{C}$ fail, i.e.,

$$\Delta^{\mathsf{A}}(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}, \mathbf{R}) \le \nu^{\mathsf{A}}(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}, \overline{C_q}) \ .$$

Below we explain how to use the adversary A to construct[5] a *non-adaptive* adversary A_{na} such that

$$\nu^{\mathsf{A}}(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}, \overline{C_q}) = \nu^{\mathsf{A}_{\mathsf{na}}}(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}, \overline{C_q}) \ . \tag{6.2}$$

[5] One could use a lemma from the random system framework [46] in the spirit of Lemma 6(ii) to switch to non-adaptivity. We prefer to spell out the actual construction to emphasize the uniformity of our reduction.

A_{na} simply runs A and responds to all its fresh queries by fresh random values, while answering repeated queries consistently. In the end, A_{na} (non-adaptively) asks all the queries that A asked during this simulated interaction. The equation (6.2) follows from the fact that the simulation for A is perfect as long as its queries do not violate $\mathcal{C}$. Since $\mathcal{C}$ is defined on $\mathsf{Casc}^{\mathsf{f}}_{K_1}$ and A_{na} is non-adaptive, we additionally have

$$\nu^{\mathsf{A}_{\mathsf{na}}}(\mathsf{Casc}^{\mathsf{f}}_{K_1} \triangleright \mathbf{r}, \overline{C_q}) = \nu^{\mathsf{A}_{\mathsf{na}}}(\mathsf{Casc}^{\mathsf{f}}_{K_1}, \overline{C_q}) .$$

Next, for A_{na} we can construct another non-adaptive adversary A_{pf} that violates the condition $\mathcal{C}$ (i.e., creates a collision in the outputs of $\mathsf{Casc}^{\mathsf{f}}_{K_1}$) with at least the same probability as A_{na}, but all its queries are *prefix-free*. This can be done, for example, by simply appending an additional block to all queries asked by A_{na}, such that this block does not appear in the original queries. Hence we have

$$\nu^{\mathsf{A}_{\mathsf{na}}}(\mathsf{Casc}^{\mathsf{f}}_{K_1}, \overline{C_q}) \leq \nu^{\mathsf{A}_{\mathsf{pf}}}(\mathsf{Casc}^{\mathsf{f}}_{K_1}, \overline{C_q})$$

for a non-adaptive adversary A_{pf} asking prefix-free queries of length at most $\ell + 1$.

Finally, consider the non-adaptive adversary A^* that simply asks the same prefix-free queries as A_{pf} and then outputs 1 if and only if the responses to these queries contain a collision. Then A^* interacting with $\mathsf{Casc}^{\mathsf{f}}_{K_1}$ outputs 1 with probability $\nu^{\mathsf{A}_{\mathsf{pf}}}(\mathsf{Casc}^{\mathsf{f}}_{K_1}, \overline{C_q})$, while in an interaction with $\mathbf{R}$ it outputs 1 with probability at most $q^2/2^c$ via the well-known birthday bound. Hence, by the definition of $\Delta^{\mathsf{A}^*}(\mathsf{Casc}^{\mathsf{f}}_{K_1}, \mathbf{R})$, we have

$$\nu^{\mathsf{A}_{\mathsf{pf}}}(\mathsf{Casc}^{\mathsf{f}}_{K_1}, \overline{C_q}) \leq \Delta^{\mathsf{A}^*}(\mathsf{Casc}^{\mathsf{f}}_{K_1}, \mathbf{R}) + \frac{q^2}{2^c} .$$

Since A^* is non-adaptive and prefix-free, we can now employ the reduction T guaranteed by Proposition 3 to obtain an NA-PRF adversary $\mathsf{T}^{\mathsf{A}^*}$ against f such that

$$\Delta^{\mathsf{A}^*}(\mathsf{Casc}^{\mathsf{f}}_{K_1}, \mathbf{R}) \leq (\ell + 1)q \cdot \Delta^{\mathsf{T}^{\mathsf{A}^*}}(\mathsf{f}, \mathbf{r}) .$$

Putting $\mathsf{T}^{\mathsf{A}}_2 := \mathsf{T}^{\mathsf{A}^*}$ hence concludes the proof of Theorem 3. □

6.3.2 Matching Attacks

We now argue that the bound obtained in Theorem 3 is essentially tight. First, we show that the term $\ell q \varepsilon_{\mathsf{na}}$ is unavoidable (up to a constant factor) by constructing a particular compression function f, which is an $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF, yet there is a simple attack against the PRF-security of $\mathsf{NMAC}^{\mathsf{f}}$ achieving advantage roughly $\ell q \varepsilon_{\mathsf{na}}$.

Proposition 4. *Let b, c, ℓ be positive integers such that $b \geq c$, let $\varepsilon_{\mathsf{na}} \in (0, 1)$, and moreover, assume that pseudo-random functions exist. Then there exists a function $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ and an adversary A against $\mathsf{NMAC}^{\mathsf{f}}$ such that for any q that satisfies $\varepsilon_{\mathsf{na}} = \omega(q^2 2^{-b}, 2^{-c})$, we have:*

- f *is $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF;*
- *the adversary A, when asking q queries of length ℓ blocks each, runs in time $\tilde{O}(\ell q)$ and achieves distinguishing advantage*

$$\Delta^{\mathsf{A}}(\mathsf{NMAC}^{\mathsf{f}}_{K}, \mathbf{R}) = \Theta(\ell q \varepsilon_{\mathsf{na}}) .$$

In particular, $\mathsf{NMAC}^{\mathsf{f}}$ is not an $(o(\ell q \varepsilon_{\mathsf{na}}), \tilde{O}(\ell q), q, \ell)$-secure PRF.

Proof sketch. Here we only describe the high-level idea for constructing f and A and defer the discussion of the technical obstacles in implementing this idea to Appendix 6.5.2.

Roughly speaking, we construct an $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF f that behaves pseudorandomly for all keys except for a small, $\varepsilon_{\mathsf{na}}/2$-fraction of them. We denote the set of these keys by $\mathcal{K}$ and refer to them as the *weak keys*. Under any weak key k, the function $\mathsf{f}(k, \cdot)$ outputs some constant value $w \in \mathcal{K}$ irrespective of its input.

To attack the NA-PRF security of $\mathsf{NMAC}^{\mathsf{f}}_{K=(K_1,K_2)}$, consider a pair of messages M_1, M_2 chosen by sampling $M \leftarrow \{0,1\}^{b(\ell-1)}$ at random and then setting $M_1 = M \| x_1$ and $M_2 = M \| x_2$ for some distinct blocks $x_1, x_2 \in \{0,1\}^b$. If some of the $\ell - 1$ intermediate values in the evaluation of the inner function $\mathsf{Casc}^{\mathsf{f}}(K_1, M)$ is in $\mathcal{K}$, then all following intermediate values are w, and in particular we have $\mathsf{Casc}^{\mathsf{f}}(K_1, M_i) = w$ for both $i \in \{1,2\}$, and hence also $\mathsf{NMAC}^{\mathsf{f}}(K, M_1) = \mathsf{NMAC}^{\mathsf{f}}(K, M_2) = \mathsf{f}_{K_2}(w)$. This implies that it is much more likely to get a collision for a pair of messages as described above for $\mathsf{NMAC}^{\mathsf{f}}_K$ than for $\mathbf{R}$. Our adversary A simply choses $q/2$ message pairs at random as above, and it outputs 1 if it observes a collision for at least one of those pairs. As there are $q/2$ message pairs, each of length ℓ, we have a total of $\ell q/2$ possibilities to "hit" a weak key, each having probability $\varepsilon_{\mathsf{na}}$. By the union bound this gives us a total probability of $\Theta(\ell q \varepsilon_{\mathsf{na}})$ for observing a collision when querying $\mathsf{NMAC}^{\mathsf{f}}_K$. On the other hand the probability of observing a colliding pair in $\mathbf{R}$ is only $O(q/2^c)$. □

We emphasize that the above attack only uses messages of one particular length and hence works equally well also if the hash function applies some length-dependent padding such as the MD-strengthening.

We now consider the tightness of the bound in Theorem 3 when $\varepsilon \gg \ell q \varepsilon_{\mathsf{na}}$ is the dominating term. This is the case when the best adaptive attack against f is by more than a factor ℓq better than any non-adaptive attack.

In [55] a pair $\mathsf{g}_1, \mathsf{g}_2$ of PRFs is constructed such that g_1 and g_2 are $\varepsilon_{\mathsf{na}}$-secure *non-adaptive* PRFs for some negligible $\varepsilon_{\mathsf{na}}$, and the serial composition $\mathsf{g}_1 \triangleright \mathsf{g}_2$ with independent keys can be broken by an *adaptive* attack (in a constant number of queries) with advantage almost 1.[6] From such $\mathsf{g}_1, \mathsf{g}_2$ we can get a single PRF f which is an $\varepsilon_{\mathsf{na}}$-secure NA-PRF for a negligible $\varepsilon_{\mathsf{na}}$, an ε-secure PRF for any ε of our choice, and where $\mathsf{f} \triangleright \mathsf{f}$ is not $\Theta(\varepsilon^2)$-secure, by setting $\mathsf{f} := \mathsf{g}_1$ and $\mathsf{f} := \mathsf{g}_2$ with probability $\varepsilon/2$, respectively, and some strong standard PRF with probability $1 - \varepsilon$ (over the choice of the key). We now observe that $\mathsf{NMAC}^{\mathsf{f}}_K$ computed on single-block messages is simply a cascade of two f's with independent keys. Thus, when using the above ε-secure PRF f, we can break $\mathsf{NMAC}^{\mathsf{f}}_K$ with advantage $\Theta(\varepsilon^2)$. This shows that the ε term in Theorem 3 is necessary if ε is constant as then $\Theta(\varepsilon) = \Theta(\varepsilon^2) = \Theta(1)$. We conjecture that $\Theta(\varepsilon^2)$ is the correct value, and the ε term in the lower bound can be improved to $\Theta(\varepsilon^2)$ using security amplification techniques along the lines of [48, 60].

6.4 PRF-Security of the NI Construction

In this section we analyze the PRF-security of the constructions NI^{h} and $\mathsf{NI2}^{\mathsf{h}}$ under the assumption that the keyed compression function h is a PRF (when keyed via its k-bit

[6] The NA-PRF security of this construction relies on the DDH assumption, [16] construct such a PRF under the weaker assumption that "uniform transcript key-agreement" exists, and this assumption is necessary [56].

input).

Recall that $d'(n)$ denotes the maximum, over all positive integers $n' \leq n$, of the number of positive divisors of n'; i.e., $d'(n) := \max_{n' \in \{1,\ldots,n\}} |\{d \in \mathbb{N} : d \mid n'\}|$.

Theorem 4. *If* $\mathsf{h}\colon \{0,1\}^k \times \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ *is an* (ε_1, t, q)*-secure PRF and an* $(\varepsilon_2, t, \ell q)$*-secure PRF, then* NI^{h} *is an* $(\varepsilon', t', q, \ell)$*-secure PRF with*

$$\varepsilon' = \varepsilon_1 + \varepsilon_2 + \frac{q^2}{2^c} \cdot \left(\ell + \frac{64\ell^4}{2^c}\right) \qquad \text{and} \qquad t = t' + \tilde{O}(\ell q)\ ,$$

and $\mathsf{NI2}^{\mathsf{h}}$ *is an* $(\varepsilon'', t'', q, \ell)$*-secure PRF with*

$$\varepsilon'' = \varepsilon_1 + \varepsilon_2 + \frac{q^2}{2^c} \cdot \left(\ell \cdot d'(\ell) + \frac{64\ell^4}{2^c}\right) \qquad \text{and} \qquad t = t'' + \tilde{O}(\ell q)\ .$$

Proof. We first prove Theorem 4 for the case of $\mathsf{NI2}^{\mathsf{h}}$ and then derive the simpler case NI^{h} from it. The proof proceeds in four consecutive steps. First, we use the PRF-security of h to replace it by an ideal compression function, making the rest of our analysis information-theoretic. Second, we observe that the resulting system behaves identically to $\mathbf{R}$ as long as no non-trivial collision occurs in the outputs of the initial cascade. Third, we reduce estimating the probability of such a collision to a counting problem of upper-bounding the number of graphs satisfying certain properties (modeling the computation of the cascade). Finally, we give a bound on the number of these graphs, hence concluding the argument.

From a PRF to a Random Function. Let A be a PRF-adversary against $\mathsf{NI2}^{\mathsf{h}}$ running in time t and asking q queries, each of length at most ℓ blocks. To simplify the notation let $\mathbf{0} := 0^c$. By a standard argument as in the proof of Theorem 3, we have

$$\Delta^{\mathsf{A}}(\mathsf{NI2}^{\mathsf{h}}_K, \mathbf{R}) = \Delta^{\mathsf{A}}\left(\mathsf{ZCasc}_{\mathbf{0}}^{\mathsf{h}_{K_1}} \triangleright \mathsf{h}_{K_2}, \mathbf{R}\right) \leq \varepsilon_1 + \varepsilon_2 + \Delta^{\mathsf{A}}\left(\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2, \mathbf{R}\right) \tag{6.3}$$

where $K = (K_1, K_2) \leftarrow (\{0,1\}^k)^2$ is a uniformly random key and $\mathbf{f}_1$ and $\mathbf{f}_2$ are two independent ideal compression functions. Interestingly, the system $\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2$ is very similar to NMAC with an ideal compression function and keys fixed to zeroes.

Bound via Collision Probability. Let $\mathsf{CColl}(\ell)$ denote the probability that a random choice of the compression function $\mathbf{f}_1$ results in a collision in $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$, maximized over the choice of the two distinct inputs to the cascade m_1, m_2 consisting of at most ℓ blocks each. (Note that this implies a collision also for $\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1}$.) Formally, for uniformly random $\mathbf{f}_1 \leftarrow \mathcal{F}(c+b, c)$ we define

$$\mathsf{CColl}(\ell) := \max_{\substack{m_1 \neq m_2 \\ |m_1|, |m_2| \leq \ell b}} \mathsf{Pr}^{\mathbf{f}_1}\left[\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}(m_1) = \mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}(m_2)\right]\ . \tag{6.4}$$

In the experiment where A interacts with $\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2$, let E_i denote the event that during the first i queries to $\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2$, for any two distinct queries M and M' the values $\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1}(M)$ and $\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1}(M')$ (inputs to the final $\mathbf{f}_2$-call) were also distinct. As long as the monotone condition $\mathcal{E} = E_0, E_1, \ldots$ remains satisfied, the responses of $\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2$ to distinct queries are clearly independent, uniformly random values thanks to $\mathbf{f}_2$. Hence, we have $(\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2)|\mathcal{E} \equiv \mathbf{R}$ and $\mathsf{p}^{\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2}_{E_i | X^i Y^{i-1} E_{i-1}} = \mathsf{p}^{\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2}_{E_i | X^i E_{i-1}}$ and can therefore consecutively apply Lemma 6(i), Lemma 6(ii), and finally the union bound to get

$$\Delta^{\mathsf{A}}(\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2, \mathbf{R}) \leq \nu(\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2, \overline{E_q}) \leq \mu(\mathsf{ZCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2, \overline{E_q}) \leq q^2 \cdot \mathsf{CColl}(\ell)\ . \tag{6.5}$$

Graph-Based Representation of Casc. The probability $\mathsf{CColl}(\ell)$ could trivially be upper-bounded by $O(\ell^2/2^c)$ using a union-bound argument, achieving a non-trivial and significantly better bound on $\mathsf{CColl}(\ell)$ is the central part of our proof. To this end, we use an approach inspired by [11] and represent the computation of $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$ on various inputs by directed graphs.

Let m_1 and m_2 be two distinct messages that can be parsed into b-bit blocks as $m_i = m_i^1\|\cdots\|m_i^{\ell_i}$ for some $\ell_1, \ell_2 \leq \ell$, and let $\Lambda := \ell_1 + \ell_2$. For convenience, we use the notation $m^{(i)}$ as a reference to the block m_1^i if $i \leq \ell_1$, otherwise it denotes the block $m_2^{i-\ell_1}$. For any fixed compression function $f \in \mathcal{F}(c+b, c)$ and a pair of such messages $\mathcal{M} = (m_1, m_2)$, we define the *structure graph* $G_f^{\mathcal{M}}$ to be the triple $G_f^{\mathcal{M}} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$, such that:

- $(\mathcal{V}, \mathcal{E})$ is a directed graph. To describe it, let
$$s_i := \begin{cases} \mathbf{0} & \text{for } i = 0 \\ f(s_{i-1}, m_1^i) & \text{for } 1 \leq i \leq \ell_1 \\ f(\mathbf{0}, m_2^1) & \text{for } i = \ell_1 + 1 \\ f(s_{i-1}, m_2^{i-\ell_1}) & \text{for } \ell_1 + 2 \leq i \leq \Lambda \end{cases} \tag{6.6}$$
and consider the mappings $[\cdot]_G$ and $[\cdot]'_G$ defined on $\{0, \ldots, \Lambda\}$ such that $[i]_G := \min\{j : s_i = s_j\}$ (so $[i]_G = i$ if and only if s_i is "fresh") and $[i]'_G := [i]_G$ for $i \neq \ell_1$, while $[\ell_1]'_G := 0$. Now we let
$$\mathcal{V} := \{[i]_G : 0 \leq i \leq \Lambda\} \quad \text{and} \quad \mathcal{E} := \{([i-1]'_G, [i]_G) : 1 \leq i \leq \Lambda\}\ .$$
- $\mathcal{L}\colon \mathcal{V}^2 \to \mathsf{Pow}(\{0,1\}^b)$ is a labeling function that labels every edge $(u, v) \in \mathcal{E}$ with the set $\{m^{(i)} : [i-1]'_G = u \wedge [i]_G = v\}$ and every pair of vertices that do not form an edge with the empty set $\emptyset$ (to simplify our notation later).

Intuitively, if all the values s_i are distinct, $G_f^{\mathcal{M}}$ simply consists of two directed paths starting in the root vertex 0, representing the evaluation of $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$ on the messages m_1 and m_2 (the edges are labeled by the corresponding blocks). If some collisions among the values s_i occur, one can obtain the graph $G_f^{\mathcal{M}}$ by collapsing every pair of vertices i, j where $s_i = s_j$ into one vertex labeled $\min\{i, j\}$, as well as merging the edge labels in the natural way.

Let $\mathcal{G}(\mathcal{M}) := \{G_f^{\mathcal{M}} : f \in \mathcal{F}(c+b, c)\}$ denote the set of all structure graphs associated with the message pair $\mathcal{M}$. Note that the uniformly distributed random variable $F \leftarrow \mathcal{F}(c+b, c)$ also induces a distribution on $\mathcal{G}(\mathcal{M})$, therefore we denote by $G_F^{\mathcal{M}}$ the resulting random variable (taking on structure graphs as values). Similarly, F also induces a distribution on the values s_i defined above and we denote the resulting random variables S_i.

For a fixed structure graph $G = G_f^{\mathcal{M}}$ we denote by $G_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{L}_i)$ the graph that is obtained after processing only the first i out of Λ blocks of $\mathcal{M}$. More formally, $G_i := G_f^{\mathcal{M}'}$ where $\mathcal{M}' := (m_1^1\|\cdots\|m_1^i, \lambda)$ if $i \leq \ell_1$ and $\mathcal{M}' := (m_1, m_2^1\|\cdots\|m_2^{i-\ell_1})$ otherwise. Building on this notion, we call $\mathsf{fColl}(G)$ the *set of f-collisions* that occurred in G:
$$\mathsf{fColl}(G) := \left\{(i, [i]_G) : [i]_G < i \wedge m^{(i)} \notin \mathcal{L}_{i-1}([i-1]'_G, [i]_G)\right\}\ . \tag{6.7}$$

Informally, imagine we reveal the structure graph G step by step, i.e., by a sequence of transitions from G_{i-1} to G_i, for $i = 1, \ldots, \Lambda$. The pair $(i, [i]_G)$ belongs to $\mathsf{fColl}(G)$ (and we

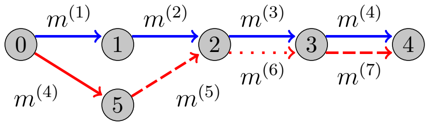


Figure 6.3: Illustration of the three cases from Lemma 7.

say that the i-th step caused an f-collision), if during this step, instead of adding a new vertex, we arrive at a vertex already visited, while not following an existing edge already labeled with $m^{(i)}$ (i.e., not repeating a step we have made before).

Properties of Structure Graphs. We first upper-bound the probability of $G_F^{\mathcal{M}}$ taking the form of any particular fixed structure graph $g \in \mathcal{G}(\mathcal{M})$. The following result and its proof is inspired by Lemma 8 from [11].

Lemma 7. *Let $F \leftarrow \mathcal{F}(c+b, c)$ be chosen uniformly at random. For a fixed graph $g \in \mathcal{G}(\mathcal{M})$ we have*

$$\mathsf{Pr}^F \left[G_F^{\mathcal{M}} = g \right] \leq 2^{-c \cdot |\mathsf{fColl}(g)|} .$$

Proof of Lemma 7. Let $\mathcal{M} = \{m_1, m_2\}$, $\Lambda = \ell_1 + \ell_2$ and let $m^{(i)}$ denote the i-th block of $m_1 \| m_2$ as before. First, we introduce the notion of consistency. Assume we sample $F \leftarrow \mathcal{F}(c+b, c)$ and the values $S_1, \ldots, S_\Lambda$ belonging to $G = G_F^{\mathcal{M}}$ are revealed to us stepwise. (Recall that S_i is the random variable representing the chaining variable of the cascade defined in (6.6) and determined by the choice of F. In turn, the values $S_1, \ldots, S_\Lambda$ completely determine the shape of the structure graph G.) We say that G is *consistent* with the given graph g after step $i \leq \Lambda$, denoted Cons_i, if the structure graphs G_i and g_i are equal as triples $(\mathcal{V}, \mathcal{E}, \mathcal{L})$ (as before, G_i denotes the part of graph G obtained after the first i blocks are processed, and g_i is defined analogously from g).

Let us assume that Cons_i is true for some i and then bound the probability $\mathsf{Pr}[\mathsf{Cons}_{i+1} | \mathsf{Cons}_i]$. To this end, we inspect the $(i+1)$-th step in g where there are the following 3 possibilities how the next edge corresponding to $m^{(i+1)}$ might look (see also Fig. 6.3):

Fresh: It arrives at a new vertex not present in g_i (i.e., $[i+1]_g = i+1$).

Determined: It follows an already existing edge (i.e., $[i+1]_g \leq i$ and $m^{(i+1)}$ is already in the label set of the edge $([i]_g, [i+1]_g)$ in g_i).

Collision: It causes an f-collision (i.e., $[i+1]_g \leq i$ and $m^{(i+1)}$ is not in the label set of the edge $([i]_g, [i+1]_g)$ in g_i). In this case, G_{i+1} will stay consistent if and only if its $(i+1)$-th edge lands on precisely the same vertex as in g_{i+1}, in other words, if $S_{i+1} = s_{i+1}$. The probability of this event (conditioned on Cons_i) is 2^{-c}, as S_{i+1} is uniformly random over $\{0,1\}^n$ and not determined in the first i steps.

Since the third case occurs exactly $|\mathsf{fColl}(g)|$ times, if we trivially upper-bound the probabilities $\mathsf{Pr}[\mathsf{Cons}_{i+1} | \mathsf{Cons}_i]$ in the other two cases by 1, we obtain the final bound $\mathsf{Pr}[G = g] = \mathsf{Pr}[\mathsf{Cons}_\Lambda] \leq 2^{-c \cdot |\mathsf{fColl}(g)|}$ as desired. □

Using Lemma 7, it is easy to see that the event that at least two f-collisions occur in G is highly unlikely.

Lemma 8. *Let $F \leftarrow \mathcal{F}(c+b,c)$ be chosen uniformly at random. Then*

$$\mathsf{Pr}^F\left[\left|\mathsf{fColl}\left(G_F^{\mathcal{M}}\right)\right| \geq 2\right] \leq \frac{4\Lambda^4}{2^{2c}}\,.$$

Proof of Lemma 8. Denote by $\mathcal{G}^r(\mathcal{M}) := \{G \in \mathcal{G}(\mathcal{M}) : |\mathsf{fColl}(G)| = r\}$ the set of all structure graphs for $\mathcal{M}$ containing exactly r f-collisions. Then (using Lemma 7 in the last step) we have

$$\begin{aligned}\mathsf{Pr}\left[\left|\mathsf{fColl}\left(G_F^{\mathcal{M}}\right)\right| \geq 2\right] &= \sum_{r=2}^{\infty}\mathsf{Pr}\left[\left|\mathsf{fColl}\left(G_F^{\mathcal{M}}\right)\right| = r\right]\\ &= \sum_{r=2}^{\infty}\sum_{g\in\mathcal{G}^r(\mathcal{M})}\mathsf{Pr}\left[G_F^{\mathcal{M}} = g\right]\\ &\leq \sum_{r=2}^{\infty}\frac{|\mathcal{G}^r(\mathcal{M})|}{(2^c)^r}\,.\end{aligned}$$

Since one can verify that any $G \in \mathcal{G}(\mathcal{M})$ is completely determined by the set of its f-collisions $\mathsf{fColl}(G) \subseteq \{(i,j) : 0 \leq j < i \leq \Lambda\}$ and the latter set has $\Lambda(\Lambda+1)/2$ elements, we have $|\mathcal{G}^r(\mathcal{M})| \leq (\Lambda(\Lambda+1)/2)^r$ and hence

$$\mathsf{Pr}\left[\left|\mathsf{fColl}\left(G_F^{\mathcal{M}}\right)\right| \geq 2\right] \leq \sum_{r=2}^{\infty}\left(\frac{\Lambda(\Lambda+1)}{2\cdot 2^c}\right)^r \leq \frac{4\Lambda^4}{2^{2c}}\,.$$

In the last step we used that $1 \leq \Lambda \leq 2^{c/2}$ and $c \geq 2$ which can be safely assumed, since otherwise the statement of the lemma is trivially true (as 1 upper-bounds any probability). □

From Collision Probability to Counting Graphs. We can now proceed to upper-bounding the value $\mathsf{CColl}(\ell)$. Let $\mathcal{M} := (m_1, m_2)$ be the two distinct messages of length at most ℓ blocks that maximize the probability

$$\mathsf{CColl}(\ell) := \max_{m_1\neq m_2}\mathsf{Pr}^F\left[\mathsf{Casc}_{\mathbf{0}}^F(m_1) = \mathsf{Casc}_{\mathbf{0}}^F(m_2)\right].$$

For $j \in \{1,2\}$ let V_j^i be the random variable denoting the i-th vertex (counting from 0) in the path corresponding to m_j in $G_F^{\mathcal{M}}$ (randomness taken over the uniform choice of F). Formally, $V_1^i := [i]_G$ and $V_2^i := [\ell_1 + i]'_G$. We also refer to the path $V_j^0, \ldots, V_j^{\ell_j}$ as the m_j*-path*. Using this notation, we have $\mathsf{CColl}(\ell) = \mathsf{Pr}[V_1^{\ell_1} = V_2^{\ell_2}]$. Since $m_1 \neq m_2$, $V_1^{\ell_1} = V_2^{\ell_2}$ cannot occur without any f-collision, hence we can split $\mathsf{CColl}(\ell)$ into

$$\mathsf{Pr}\left[V_1^{\ell_1} = V_2^{\ell_2} \wedge |\mathsf{fColl}(G_F^{\mathcal{M}})| = 1\right] + \mathsf{Pr}\left[V_1^{\ell_1} = V_2^{\ell_2} \wedge |\mathsf{fColl}(G_F^{\mathcal{M}})| \geq 2\right]\,. \tag{6.8}$$

The latter probability can be readily upper-bounded by $4\Lambda^4/2^{2c}$ using Lemma 8. As for the former, let us denote by $\mathcal{H}(\mathcal{M})$ the set

$$\mathcal{H}(\mathcal{M}) := \left\{G \in \mathcal{G}^1(\mathcal{M}) : V_1^{\ell_1} = V_2^{\ell_2}\right\}$$

of structure graphs for $\mathcal{M}$ that contain exactly one f-collision and where the vertices $V_1^{\ell_1}$ and $V_2^{\ell_2}$ coincide. The first term in (6.8) can then be upper-bounded by $|\mathcal{H}(\mathcal{M})|/2^c$ using Lemma 7, hence it remains to bound the size of the set $\mathcal{H}(\mathcal{M})$.

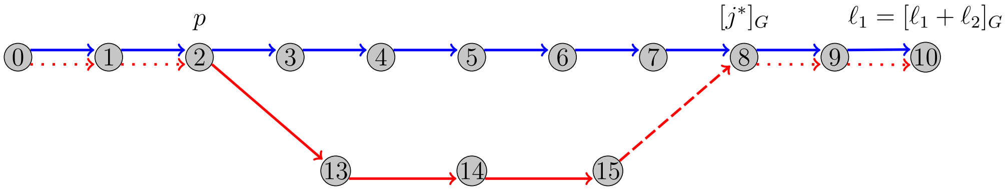


Figure 6.4: A sample graph from the set $\mathcal{H}_1$ in the proof of Lemma 9, with $p = 2$ and $j^* = 16$.

Counting the Structure Graphs. We give such a bound in the following lemma. Recall that $d'(n)$ denotes the maximum, over all positive integers $n' \leq n$, of the number of positive divisors of n'; i.e., $d'(n) := \max_{n' \in \{1,\ldots,n\}} |\{d \in \mathbb{N} : d \mid n'\}|$.

Lemma 9. *For two distinct messages $\mathcal{M} = \{m_1, m_2\}$ each of length at most ℓ blocks we have $|\mathcal{H}(\mathcal{M})| \leq \ell d'(\ell)$. If the messages in $\mathcal{M}$ are of the same length then we have $|\mathcal{H}(\mathcal{M})| \leq \ell$.*

Proof of Lemma 9. Let us first consider the general case where we allow the messages m_1 and m_2 to have different lengths, let us denote them by ℓ_1 and ℓ_2 as before. Without loss of generality let us assume that $\ell_1 \geq \ell_2$. We split the set $\mathcal{H}(\mathcal{M})$ into two partitions: Let $\mathcal{H}_1$ contain all the structure graphs from $\mathcal{H}(\mathcal{M})$ such that the m_1-path does not contain a loop, and let $\mathcal{H}_2$ contain all the rest. Formally, $\mathcal{H}_1 := \{G \in \mathcal{H}(\mathcal{M}); \forall i \in \{1,\ldots,\ell_1\} : [i]_G = i\}$ and $\mathcal{H}_2 := \mathcal{H}(\mathcal{M}) \setminus \mathcal{H}_1$. We now upper-bound the size of both partitions in two separate claims, which together conclude the proof of the first part of Lemma 9.

Claim 1: $|\mathcal{H}_1| \leq \ell$.

Towards bounding $|\mathcal{H}_1|$, note that if m_2 is a prefix of m_1 then clearly $|\mathcal{H}_1| = 0$, therefore we assume that this is not the case. Let $m_1^1 \| \cdots \| m_1^p$ be the blocks forming the longest common prefix of m_1 and m_2; i.e., let $p \in \mathbb{N}$ be the smallest index such that $m_1^{p+1} \neq m_2^{p+1}$ (for illustration see Fig. 6.4). Since f is a function, we clearly have $V_1^i = V_2^i$ for all $i \leq p$. Let us now consider $j^* := \min\{j > \ell_1 + p : [j]_G \leq \ell_1\}$. Such a j^* is well-defined, since at least the value $\ell_1 + \ell_2$ belongs to the considered set (we have $\ell_1 + \ell_2 > \ell_1 + p$ and $[\ell_1 + \ell_2]_G = \ell_1$).

We now prove that the j^*-th edge $([j^* - 1]'_G, [j^*]_G)$ in G must create an f-collision, i.e., that $(j^*, [j^*]_G) \in \mathsf{fColl}(G)$. We have $[j^*]_G \in \mathcal{V}_{j^*-1}$ by definition of j^* and to also see that $m^{(j^*)} \notin \mathcal{L}_{j^*-1}([j^* - 1]'_G, [j^*]_G)$ we consider two cases:

1. If $[j^*]_G \geq 1$ and $[j^*]_G - 1 = [j^* - 1]'_G$ (the vertices directly preceding the vertex $V_1^{[j^*]_G}$ on m_1-path and m_2-path coincide), then we must have $j^* = p + 1$, otherwise this would contradict the minimality of j^*. However, this implies that $m^{([j^*]_G)} \neq m^{(j^*)}$ (as otherwise the common prefix would be longer than p blocks) and hence $m^{(j^*)} \notin \mathcal{L}_{j^*-1}([j^* - 1]'_G, [j^*]_G) = \{m^{([j^*]_G)}\}$.

2. On the other hand, if $[j^*]_G - 1 \neq [j^* - 1]'_G$, then we claim that there was no edge $([j^* - 1]'_G, [j^*]_G)$ in G_{j^*-1} and hence $m^{(j^*)} \notin \mathcal{L}_{j^*-1}([j^* - 1]'_G, [j^*]_G) = \emptyset$. Indeed, the only edge leading into the vertex $[j^*]_G$ in G_{j^*-1} can be from $[j^*]_G - 1$, as anything else would contradict either the absence of cycles within the m_1-path, or the minimality of j^*.

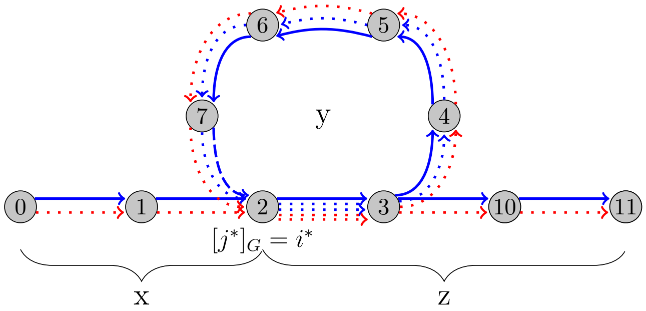


Figure 6.5: A sample graph from the set $\mathcal{H}_2$ in the proof of Lemma 9, with $i^* = 2$ and $j^* = 8$.

Given the j^*-th edge causes an f-collision and $|\mathsf{fColl}(G)| = 1$, no f-collision in G occurs beyond the j^*-th edge. However, we have $[\ell_1]_G = [\ell_1 + \ell_2]_G$ and to achieve this without any additional collision, clearly, we need that $m^{([j^*]_G+1)} \| \cdots \| m^{(\ell_1)} = m^{(j^*+1)} \| \cdots \| m^{(\ell_1+\ell_2)}$, i.e., the suffixes of m_1 and m_2 after the collision are the same. This, however, implies that the value j^* completely determines the structure graph within $\mathcal{H}_1$ and hence we arrive at $|\mathcal{H}_1| \leq \ell$.

Claim 2: $|\mathcal{H}_2| \leq \ell \cdot d'(\ell)$.

For this part, let $j^* := \min\{j : [j]_G < j\}$ and $i^* := [j^*]_G$, where such a $j^* \leq \ell_1$ exists by definition of $\mathcal{H}_2$. Moreover, it creates an f-collision (i.e., $(j^*, i^*) \in \mathsf{fColl}(G)$) by an argument similar to the one from Claim 1. We now split m_1 into $x := m_1^1 \| \cdots \| m_1^{i^*}$, $y := m_1^{i^*+1} \| \cdots \| m_1^{j^*}$ and some z that is chosen to be the shortest string possible such that $m_1 = x \| y^k \| z$ holds for some $k \geq 1$ (note that such z always exists and is unique, possibly empty). This situation is illustrated in Fig. 6.5.

We claim that in any $G \in \mathcal{H}_2$ the m_2-path is a subgraph of the m_1-path (ignoring the labels for now). Indeed, if the m_2-path contained any edges not contained in the m_1-path, then (since $V_1^{\ell_1} = V_2^{\ell_2}$) the last such “outlying” edge would create an f-collision. To see this, observe that since this is the last edge not in the path of m_1 its end vertex will be contained in the path of both messages, which causes an f-collision when this edge is added (see (6.7)). However, the m_1-path already created one f-collision and hence creating another one would violate the definition of $\mathcal{H}(\mathcal{M})$.

Moreover, for the same reason the m_2-path cannot introduce new labels to the edges in m_1-path, as this would cause another f-collision. This implies that m_2 has to be of the form $m_2 = x \| y^{k'} \| z$ for some $k' < k$. To achieve this, the number of blocks in y (i.e., $j^* - i^*$) must divide $\ell_1 - \ell_2$.

For any fixed $\mathcal{M}$, a structure graph in $\mathcal{H}_2$ is fully determined by the choice of $j^* \in \{1, \ldots, \ell_1\}$ and $i^* \in \{0, \ldots, j^* - 1\}$, such that $(j^* - i^*) \mid \ell_1 - \ell_2$. There are at most ℓ ways to choose such a j^* and at most $d'(\ell)$ ways to choose a consistent i^*. Consequently, we obtain $|\mathcal{H}_2| \leq \ell \cdot d'(\ell)$, which concludes the proof for the case of distinct-length messages.

For the second part of the claim it now suffices to observe that if $|m_1| = |m_2|$ then $|\mathcal{H}_2| = 0$. This is because in $\mathcal{H}_2$ the m_1-path already contains an f-collision, and since only one such f-collision is allowed to occur, the only way to achieve $V_1^{\ell_1} = V_2^{\ell_2}$ would hence be if $m_1 = m_2$. This however contradicts the assumption that the messages are distinct. □

In Appendix 6.5.3 we also show that Lemma 9 is tight, and discuss the implications for the tightness of Theorem 4.

Finally, combining the equations (6.3), (6.5), (6.8), and the bounds obtained in Lemma 8 and Lemma 9, we get

$$\Delta^{\mathsf{A}}(\mathsf{NI2}^{\mathsf{h}}_{K}, \mathbf{R}) \leq \varepsilon_1 + \varepsilon_2 + q^2 \cdot \left(\frac{\ell \cdot d'(\ell)}{2^c} + \frac{4\Lambda^4}{2^{2c}}\right) \leq \varepsilon_1 + \varepsilon_2 + \frac{q^2}{2^c} \cdot \left(\ell \cdot d'(\ell) + \frac{64\ell^4}{2^c}\right)$$

and conclude the proof of Theorem 4 for $\mathsf{NI2}^{\mathsf{h}}$.

The case of NI is handled in the same way as $\mathsf{NI2}$, with the only difference being that it contains $\mathsf{LenCasc}$ instead of ZCasc. Hence, to imply a collision for $\mathsf{LenCasc}$, we require the messages m_1 and m_2 in the definition of $\mathsf{CColl}(\ell)$ to be of the same length. This leads to the use of the second part of Lemma 9 that assumes equal-length messages, arriving at the claimed bound. □

6.5 Appendix

6.5.1 Non-Adaptive Security of the Cascade

Here we prove Proposition 3 that states the PRF-security of the construction $\mathsf{Casc}^{\mathsf{f}}$ against non-adaptive prefix-free adversaries, assuming that f itself is a non-adaptively secure PRF. Our argument follows the proof for the adaptive case in [8] with minor modifications and we include it here for completeness.

Given a compression function $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ and a tuple of independent random keys $\overline{K} = (K_1, \ldots, K_q) \in (\{0,1\}^c)^q$, let $\mathsf{qf}_{\overline{K}} = (\mathsf{f}_{K_1}, \ldots, \mathsf{f}_{K_q})$ denote the q-tuple of oracles providing access to q copies of f, each one being assigned a different key from $\overline{K}$. Moreover, let $\mathbf{qr} = (\mathbf{r}_1, \ldots, \mathbf{r}_q)$ denote the q-tuple of independent, uniformly random functions $\mathbf{r}_i\colon \{0,1\}^b \to \{0,1\}^c$. Following [8], we say that f is (ε, t, q)-NA-PRF^q-secure, if for any non-adaptive adversary A running in time t and asking at most q queries, we have $\Delta^{\mathsf{A}}(\mathsf{qf}_{\overline{K}}, \mathbf{qr}) \leq \varepsilon$.

Proposition 3 (restated). *Let $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ be a compression function. There exists an explicit reduction T (described in the proof) such that for any $(\varepsilon', t', q, \ell)$-NA-PF-PRF adversary A against $\mathsf{Casc}^{\mathsf{f}}$, T^{A} is an $(\varepsilon_{\mathsf{na}}, t, q)$-NA-PRF adversary against f such that*

$$\varepsilon' \leq \ell q \varepsilon_{\mathsf{na}} \qquad \text{and} \qquad t = t' + \tilde{O}(\ell q)\ .$$

Proof. The proof consists of two consecutive reductions. First, out of an assumed attacker against the NA-PF-PRF security of $\mathsf{Casc}^{\mathsf{f}}$ we construct an attacker against the NA-PRF^q security of f. Second, we use the latter to construct an attacker against the NA-PRF-security of f. In each of these two steps the success probabilities of the two attackers are related by a hybrid argument. We describe and analyze each of these two steps in a separate lemma below.

Lemma 10. *There exists an explicit reduction T_1 (described in the proof) such that for any non-adaptive adversary A_1 against the NA-PF-PRF security of $\mathsf{Casc}^{\mathsf{f}}$, running in time t' and asking q prefix-free queries of length at most ℓ blocks each, $\mathsf{A}_2 := \mathsf{T}_1^{\mathsf{A}_1}$ is a non-adaptive adversary against the NA-PRF^q-security of f running in time $t' + O(\ell q)$ and asking at most q queries, such that $\Delta^{\mathsf{A}_1}(\mathsf{Casc}^{\mathsf{f}}_{K}, \mathbf{R}) \leq \ell \cdot \Delta^{\mathsf{A}_2}(\mathsf{qf}_{\overline{K}}, \mathbf{qr})$.*

Proof of Lemma 10. We start by describing a sequence of adversaries $\mathsf{A}_2^{(i)}$ for $i \in \{1, \ldots, \ell\}$. Given access to oracles $(\mathsf{g}_1, \ldots, \mathsf{g}_q)$ which are either $\mathsf{qf}_{\overline{K}} = (\mathsf{f}_{K_1}, \ldots, \mathsf{f}_{K_q})$ (for independent random keys $K_1, \ldots, K_q$), or q independent random functions $\mathbf{qr} = (\mathbf{r}_1, \ldots, \mathbf{r}_q)$, $\mathsf{A}_2^{(i)}$ works as follows:

1. It runs A_1 to obtain its q non-adaptive prefix-free queries $x_1, \ldots, x_q$, each of length at most ℓ blocks (without loss of generality we assume that $x_1, \ldots, x_q \in \{0,1\}^{b*}$ are distinct). Each query x_j is parsed into blocks as $x_j = x_j^1 \| \cdots \| x_j^{\ell_j}$, where each $x_j^z \in \{0,1\}^b$.

2. The response r_j to each query x_j is determined: If $\ell_j < i$, then r_j is chosen independently and uniformly at random. Otherwise, an index $c_j \in \{1, \ldots, q\}$ is determined consecutively for all queries of length at least i in an arbitrary way, given that two queries x_j and $x_{j'}$ share the same index (i.e., $c_j = c_{j'}$) if and only if their first $i-1$ blocks are identical (i.e., $x_j^1 \| \cdots \| x_j^{i-1} = x_{j'}^1 \| \cdots \| x_{j'}^{i-1}$). The response r_j is then computed as
$$r_j \leftarrow \mathsf{Casc}^{\mathsf{f}}_{\mathsf{g}_{c_j}(x_j^i)} \left(x_j^{i+1} \| \cdots \| x_j^{\ell_j} \right) .$$
All g-values required for this computation are obtained by querying the g-oracles; note that this can be done non-adaptively. The tuple of responses $(r_1, \ldots, r_q)$ is given to A_1.

3. $\mathsf{A}_2^{(i)}$ outputs the same bit that A_1 does.

A straightforward analysis using the definition of $\mathsf{A}_2^{(i)}$ allows one to establish the following three facts:

(i) $\mathsf{A}_1(\mathsf{Casc}^{\mathsf{f}}_K) = \mathsf{A}_2^{(1)}(\mathsf{qf}_{\overline{K}})$,

(ii) $\mathsf{A}_1(\mathbf{R}) = \mathsf{A}_2^{(\ell)}(\mathbf{qr})$,

(iii) $\mathsf{A}_2^{(i+1)}(\mathsf{qf}_{\overline{K}}) = \mathsf{A}_2^{(i)}(\mathbf{qr})$ for all $i \in \{1, \ldots, \ell\}$,

where the equalities represent equal distributions of the output bits. Combining these facts, we get
$$\begin{aligned} \Delta^{\mathsf{A}_1}(\mathsf{Casc}^{\mathsf{f}}_K, \mathbf{R}) &= \left| \mathsf{Pr}[\mathsf{A}_1(\mathsf{Casc}^{\mathsf{f}}_K) = 1] - \mathsf{Pr}[\mathsf{A}_1(\mathbf{R}) = 1] \right| \\ &\overset{(i),(ii)}{=} \left| \mathsf{Pr}[\mathsf{A}_2^{(1)}(\mathsf{qf}_{\overline{K}}) = 1] - \mathsf{Pr}[\mathsf{A}_2^{(\ell)}(\mathbf{qr}) = 1] \right| \\ &\overset{(iii)}{\leq} \sum_{i=1}^{\ell} \left| \mathsf{Pr}[\mathsf{A}_2^{(i)}(\mathsf{qf}_{\overline{K}}) = 1] - \mathsf{Pr}[\mathsf{A}_2^{(i)}(\mathbf{qr}) = 1] \right| \\ &= \sum_{i=1}^{\ell} \Delta^{\mathsf{A}_2^{(i)}}(\mathsf{qf}_{\overline{K}}, \mathbf{qr}) . \end{aligned} \tag{6.9}$$

Now we define A_2 to initially choose an index $i \in \{1, \ldots, \ell\}$ uniformly at random and then act as $\mathsf{A}_2^{(i)}$. This implies
$$\Delta^{\mathsf{A}_2}(\mathsf{qf}_{\overline{K}}, \mathbf{qr}) = \frac{1}{\ell} \cdot \sum_{i=1}^{\ell} \Delta^{\mathsf{A}_2^{(i)}}(\mathsf{qf}_{\overline{K}}, \mathbf{qr})$$
and hence concludes the proof of Lemma 10. □

Lemma 11. *There exists an explicit reduction* T_2 *(described in the proof) such that for any non-adaptive adversary* A_2 *against the NA-PRF*^{q}*-security of* f*, running in time* $t' + O(\ell q)$ *and asking at most* q *queries,* $\mathsf{A}_3 := \mathsf{T}_2^{\mathsf{A}_2}$ *is a non-adaptive adversary against the NA-PRF-security of* f *running in time* $t' + O(\ell q)$ *and asking at most* q *queries, such that* $\Delta^{\mathsf{A}_2}(\mathsf{qf}_{\overline{K}}, \mathbf{qr}) \leq q \cdot \Delta^{\mathsf{A}_3}(\mathsf{f}_K, \mathbf{r})$*.*

Proof of Lemma 11. Let us again describe a sequence of adversaries $\mathsf{A}_3^{(i)}$ for $i \in \{1, \ldots, q\}$. Given access to an oracle g, which is either f_K (for an independent random key K), or an independent random function $\mathbf{r}$, $\mathsf{A}_3^{(i)}$ works as follows:

1. It runs A_2 to obtain its q non-adaptive queries $(o_1, x_1), \ldots, (o_q, x_q)$, each consisting of a pair (o, x) representing a query x to A_2's o-th oracle.

2. $\mathsf{A}_3^{(i)}$ chooses $i-1$ independent random keys $K_1, \ldots, K_{i-1} \in \{0,1\}^c$. Then, it determines the response r_j to each query (o_j, x_j) as
$$r_j \leftarrow \begin{cases} \mathsf{f}_{K_{o_j}}(x_j) & \text{if } o_j < i \\ \mathsf{g}(x_j) & \text{if } o_j = i \\ \mathbf{r}_{o_j}(x_j) & \text{if } o_j > i, \end{cases}$$
where $\mathbf{r}_{i+1}, \ldots, \mathbf{r}_q$ are independent uniformly random functions, sampled internally by $\mathsf{A}_3^{(i)}$ (using lazy sampling to maintain efficiency). All g-values required for this computation are obtained by querying the g-oracle and once again this can be done non-adaptively. The tuple of responses $(r_1, \ldots, r_q)$ is given to A_2.

3. $\mathsf{A}_3^{(i)}$ outputs the same bit that A_2 does.

This time it is easy to observe that we have

(iv) $\mathsf{A}_2(\mathsf{qf}_{\overline{K}}) = \mathsf{A}_3^{(q)}(\mathsf{f}_K)$

(v) $\mathsf{A}_2(\mathbf{qr}) = \mathsf{A}_3^{(1)}(\mathbf{r})$

(vi) $\mathsf{A}_3^{(i)}(\mathsf{f}_K) = \mathsf{A}_3^{(i+1)}(\mathbf{r})$ for all $i \in \{1, \ldots, q\}$

and hence, similarly as in (6.9), we get

$$\Delta^{\mathsf{A}_2}(\mathsf{qf}_{\overline{K}}, \mathbf{qr}) \overset{(iv),(v)}{=} \left|\mathsf{Pr}[\mathsf{A}_3^{(q)}(\mathsf{f}_K) = 1] - \mathsf{Pr}[\mathsf{A}_3^{(1)}(\mathbf{r}) = 1]\right| \overset{(vi)}{\leq} \sum_{i=1}^{q} \Delta^{\mathsf{A}_3^{(i)}}(\mathsf{f}_K, \mathbf{r}) \ . \tag{6.10}$$

Again, letting A_3 be an adversary that chooses a random index $i \in \{1, \ldots, q\}$ and then simulates $\mathsf{A}_3^{(i)}$ gives us
$$\Delta^{\mathsf{A}_3}(\mathsf{f}_K, \mathbf{r}) = \frac{1}{q} \cdot \sum_{i=1}^{q} \Delta^{\mathsf{A}_3^{(i)}}(\mathsf{f}_K, \mathbf{r}) \ ,$$
thus proving Lemma 11. □

The proof of Proposition 3 is now concluded by combining the two reductions described above. For any $(\varepsilon', t', q, \ell)$-NA-PF-PRF adversary A against $\mathsf{Casc}^{\mathsf{f}}$, we let $\mathsf{T}^{\mathsf{A}} := \mathsf{T}_2^{\mathsf{T}_1^{\mathsf{A}}}$ and observe that $\Delta^{\mathsf{A}}(\mathsf{Casc}_K^{\mathsf{f}}, \mathbf{R}) \leq \ell q \cdot \Delta^{\mathsf{T}^{\mathsf{A}}}(\mathsf{f}_K, \mathbf{r})$ while T^{A} runs in time $t' + \tilde{O}(\ell q)$ and asks at most q queries as desired. □

6.5.2 Proof of Proposition 4

In this appendix we fill in the details omitted in the sketch of the proof of Proposition 4 in Section 6.3.2.

Proposition 4 (restated)**.** *Let b, c, ℓ be positive integers such that $b \geq c$, let $\varepsilon_{\mathsf{na}} \in (0,1)$, and moreover, assume that pseudo-random functions exist. Then there exists a function $\mathsf{f}\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ and an adversary A against $\mathsf{NMAC}^{\mathsf{f}}$ such that for any q that satisfies $\varepsilon_{\mathsf{na}} = \omega(q^2 2^{-b}, 2^{-c})$, we have:*

- *f is $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF;*
- *the adversary A, when asking q queries of length ℓ blocks each, runs in time $\tilde{O}(\ell q)$ and achieves distinguishing advantage*

$$\Delta^{\mathsf{A}}(\mathsf{NMAC}^{\mathsf{f}}_K, \mathbf{R}) = \Theta(\ell q \varepsilon_{\mathsf{na}}) \ .$$

In particular, $\mathsf{NMAC}^{\mathsf{f}}$ is not an $(o(\ell q \varepsilon_{\mathsf{na}}), \tilde{O}(\ell q), q, \ell)$-secure PRF.

Proof. We start by showing how to construct the $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF f. To simplify our technical arguments later, we design f in such a way that besides having weak keys as sketched in Section 6.3, it also satisfies the additional property that for any key $k \in \{0,1\}^c$ and a uniformly distributed input $U \in \{0,1\}^b$, the value $\mathsf{f}(k, U)$ is also uniformly distributed. Having this goal in mind, we construct f starting from a pseudo-random permutation (which exists by our assumption and the result [41]). Consider any $(\varepsilon_{\mathsf{na}}/4, t, q)$-NA-secure PRP $\pi\colon \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^b$ and a set of "weak keys" $\mathcal{K} \subseteq \{0,1\}^c$ of size $2^c(\varepsilon_{\mathsf{na}}/2)$, defined as $\mathcal{K} := 0^{1-\log \varepsilon_{\mathsf{na}}} \| \{0,1\}^{c+\log \varepsilon_{\mathsf{na}}-1}$ (the set of keys where the first $1 - \log \varepsilon_{\mathsf{na}}$ bits are 0). Let $[\cdot]_c$ represent the truncation of a longer bitstring to its first c bits. We fix a value $w \in \mathcal{K}$ (say $w = 0^c$) and define f as

$$\mathsf{f}(k, x) := \begin{cases} w & \text{for } k \in \mathcal{K}, \\ [\pi(k, x)]_c & \text{for } k \notin \mathcal{K}. \end{cases}$$

Hence, f behaves as a truncated version of π except when a weak key from $\mathcal{K}$ is used, in this case $\mathsf{f}(k, \cdot)$ always outputs w. By the well-known PRF/PRP switching lemma [31] we obtain that π is also an $(\varepsilon_{\mathsf{na}}/4 + q^2/2^b, t, q)$-NA-secure PRF and by assumption $\varepsilon_{\mathsf{na}}/4 + q^2/2^b \leq \varepsilon_{\mathsf{na}}/2$. It is easy to see that this implies that also $[\pi(\cdot)]_c$ is an $(\varepsilon_{\mathsf{na}}/2, t, q)$-NA-secure PRF. By redefining $[\pi(\cdot)]_c$ on an $\varepsilon_{\mathsf{na}}/2$-fraction of the keys at most an $\varepsilon_{\mathsf{na}}/2$ term in the PRF-distinguishing advantage is lost, hence the function f is an $(\varepsilon_{\mathsf{na}}, t, q)$-NA-secure PRF.

Now consider two queries M_1, M_2 to $\mathsf{NMAC}^{\mathsf{f}}(K = (K_1, K_2), \cdot)$ which are determined by first sampling an $(\ell - 1)$-block message $M = m_1 \| \cdots \| m_{\ell-1} \in \{0,1\}^{b(\ell-1)}$ at random and then setting $M_1 = M \| x_1$ and $M_2 = M \| x_2$ for some distinct blocks $x_1, x_2 \in \{0,1\}^b$. Let $Z_0 := K_1$ and $Z_i := \mathsf{f}(Z_{i-1}, m_i)$ for $i \in \{1, \ldots, \ell - 1\}$. If any of the $\ell - 1$ intermediate values $Z_1, \ldots, Z_{\ell-1}$ in the evaluation of the inner function $\mathsf{Casc}^{\mathsf{f}}(K_1, M)$ is in $\mathcal{K}$, then $\mathsf{Casc}^{\mathsf{f}}(K_1, M_i) = w$ for both $i \in \{1, 2\}$ and hence also $\mathsf{NMAC}^{\mathsf{f}}(K, M_1) = \mathsf{NMAC}^{\mathsf{f}}(K, M_2)$. We now lower-bound the probability of this event occurring. Since M is chosen independently and uniformly at random, the construction of f from a permutation implies that each value Z_i will also be distributed uniformly at random and independently of $\mathcal{K}$, as long as $Z_{i-1} \notin \mathcal{K}$. Therefore, we obtain

$$\begin{aligned}\mathsf{Pr}^{K,M}\left[\{Z_1,\ldots,Z_{\ell-1}\} \cap \mathcal{K} \neq \emptyset\right] &= 1 - \mathsf{Pr}^{K,M}\left[\{Z_1,\ldots,Z_{\ell-1}\} \cap \mathcal{K} = \emptyset\right] \\ &= 1 - \left(\mathsf{Pr}^{K,M}\left[Z_0 \notin \mathcal{K}\right] \cdot \prod_{i=1}^{\ell-1} \mathsf{Pr}^{K,M}\left[Z_i \notin \mathcal{K} | Z_{i-1} \notin \mathcal{K}\right]\right) \\ &= 1 - \left(1 - \frac{\varepsilon_{\mathsf{na}}}{2}\right)^{\ell} \geq \ell\varepsilon_{\mathsf{na}}/4 \,.\end{aligned}$$

As explained above, this also lower-bounds the probability of a collision between $\mathsf{NMAC}^{\mathsf{f}}(K, M_1)$ and $\mathsf{NMAC}^{\mathsf{f}}(K, M_2)$.

Now, consider an adversary A that queries $\mathsf{NMAC}^{\mathsf{f}}_K$ on $q/2$ such random and independently sampled message pairs M_1, M_2 and outputs 1 if and only if it observes a collision for at least one such pair. A interacting with $\mathsf{NMAC}^{\mathsf{f}}_K$ outputs 1 with probability

$$1 - \left(1 - \frac{\ell\varepsilon_{\mathsf{na}}}{4}\right)^{q/2} \geq \frac{\ell q \varepsilon_{\mathsf{na}}}{16} = \Theta(\ell q \varepsilon_{\mathsf{na}}) \,.$$

However, in the interaction with the random function $\mathbf{R}$, A clearly outputs 1 with probability only $O(q/2^c)$. By our assumption on $\varepsilon_{\mathsf{na}}$, we get $q/2^c = o(\ell q \varepsilon_{\mathsf{na}})$ and hence also $\Delta^{\mathsf{A}}(\mathsf{NMAC}^{\mathsf{f}}_K, \mathbf{R}) = \Omega(\ell q \varepsilon_{\mathsf{na}})$ as desired. □

6.5.3 Tightness of Lemma 9 and Theorem 4

In this appendix we prove a lower bound, for a particular pair of messages $\mathcal{M} = \{m_1, m_2\}$, on the number of structure graphs that contain exactly one f-collision and where the final vertices $V_1^{\ell_1}$ and $V_2^{\ell_2}$ of their message paths coincide. As in Lemma 9 we consider both the case where the messages are required to have the same length, and the case without this requirement. Recall that $\mathcal{H}(\mathcal{M})$ denotes the set

$$\mathcal{H}(\mathcal{M}) := \left\{G \in \mathcal{G}^1(\mathcal{M}) : V_1^{\ell_1} = V_2^{\ell_2}\right\}$$

of such graphs and $d'(n) := \max_{n' \in \{1,\ldots,n\}} |\{d \in \mathbb{N} : d \mid n'\}|$. Proposition 5 below shows that Lemma 9 is tight (up to a constant factor 4).

Proposition 5. *There exist two distinct messages $\mathcal{M} = \{m_1, m_2\}$, each of length at most ℓ blocks, such that $|\mathcal{H}(\mathcal{M})| \geq \frac{\ell \cdot d'(\ell)}{4}$. Moreover, if we additionally require $|m_1| = |m_2|$ then there exist two equal-length messages $\mathcal{M} = \{m_1, m_2\}$ of length at most ℓ blocks such that $|\mathcal{H}(\mathcal{M})| \geq \ell$.*

Proof. Again, let us first consider the case where $|m_1| = |m_2|$ is *not* required. Given ℓ, let $\ell' \leq \ell/2$ be any positive integer such that $d(\ell') = d'(\ell/2)$ (it exists by the definition of $d'(\cdot)$). We choose $m_1, m_2 \in \{0,1\}^{b*}$ to be the messages consisting of $\ell/2 + \ell'$ and $\ell/2$ equal blocks 0^b, respectively. Now, we describe $\ell \cdot d'(\ell)/4$ distinct structure graphs and show that they are all in $\mathcal{H}(\mathcal{M})$, thus establishing the proof of the first part.

For every $i \in \{0, \ldots, \ell/2\}$ and every d that is a divisor of ℓ', we denote by $G_{i,d}$ the structure graph constructed as follows: Informally, the graph corresponding to m_1 starts with a path of length $i + d - 1$ edges, and the $(i + d)$-th edge returns to vertex i, hence

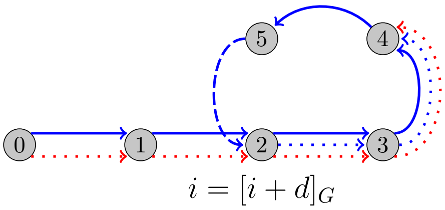


Figure 6.6: A sample graph $G_{i,d}$ for the proof of Proposition 5.

causing a collision. Note that now we have a ρ-shaped graph (where the cycle has length d), and the remaining edges of m_1 must follow the edges along the cycle in that graph. Since m_2 is a prefix of m_1, this also determines the m_2-path (see Figure 6.6 for a sample $G_{i,d}$). Formally, $G_{i,d} := (\mathcal{V}, \mathcal{E}, \mathcal{L})$ where

$$\begin{aligned} \mathcal{V} &:= \{0, \ldots, i + d - 1\}, \\ \mathcal{E} &:= \{(j-1, j) \mid 1 \le j \le i + d - 1\} \cup \{(i + d - 1, i)\} \text{ and} \\ \mathcal{L}(u, v) &:= \{0^b\} \text{ for all } (u, v) \in \mathcal{E}\ . \end{aligned}$$

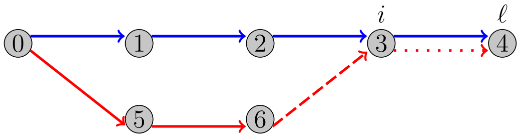
It is clear from the definition of $G_{i,d}$ that for distinct $(i, d) \neq (i', d')$ we also have $G_{i,d} \neq G_{i',d'}$. Moreover, we claim that for each (i, d) chosen as above, $G_{i,d} \in \mathcal{H}(\mathcal{M})$. To see this, observe that the m_1-path ends in the vertex $i + (\ell/2 + \ell' - i \mod d)$, while the m_2-path ends in the vertex $i + (\ell/2 - i \mod d)$. Since $d|\ell'$, this is actually the same vertex and we have $V_1^{\ell_1} = V_2^{\ell_2}$, establishing $G_{i,d} \in \mathcal{H}(\mathcal{M})$. There are $(\ell/2 + 1) \cdot d(\ell')$ ways to choose a tuple (i, d) with $i \in \{0, \ldots, \ell/2\}$ and d being a divisor of ℓ', and thus $\mathcal{H}(\mathcal{M})$ has at least $(\ell/2 + 1) \cdot d(\ell') \ge \ell/2 \cdot d'(\ell/2) \ge \ell d'(\ell)/4$ distinct elements as claimed.

For the case $|m_1| = |m_2|$, consider the messages $m_1 = 1^b 0^{b(\ell-1)}$ and $m_2 = 01^{b-1}0^{b(\ell-1)}$. These messages are both of length ℓ blocks and differ in their first blocks, while the remaining $\ell - 1$ blocks consist of zeroes in both messages. We again construct ℓ distinct structure graphs and show that they all belong to $\mathcal{H}(\mathcal{M})$.

For every $i \in \{1, \ldots, \ell\}$, we denote by $G(i)$ the structure graph constructed as follows: Informally, the subgraph corresponding to m_1 is a path of length ℓ, not containing any collision itself. Since m_2 differs from m_1 in the first block, the m_2-path will not overlap with the m_1-path as long as no f-collision occurs. In the graph $G(i)$, we let this collision happen for the i-th edge of the m_2-path, hitting the vertex i on the m_1-path. In particular, in the case $i = 1$ the collision occurs by having $V_1^1 = V_2^1$ even though the first blocks of the messages differ. See Figure 6.7 for a sample $G(i)$. Formally, $G(i) := (\mathcal{V}, \mathcal{E}, \mathcal{L})$, where

$$\begin{aligned} \mathcal{V} &:= \{0, \ldots, \ell + i - 1\}, \\ \mathcal{E} &:= \begin{cases} \{(j-1, j) \mid 1 \le j \le \ell\} & \text{if } i = 1 \\ \{(j-1, j) \mid j \in \{1, \ldots, \ell + i - 1\} \setminus \{\ell + 1\}\} & \\ \qquad \cup\{(0, \ell + 1)\} \cup \{(\ell + i - 1, i)\} & \text{if } i > 1 \end{cases} \\ \mathcal{L}(u, v) &:= \begin{cases} \{1^b, 01^{b-1}\} & \text{if } (u, v) = (0, 1) \text{ and } i = 1 \\ \{1^b\} & \text{if } (u, v) = (0, 1) \text{ and } i > 1 \\ \{0^b\} & \text{if } (u, v) \in \{(j-1, j) \mid j \in \{1, \ldots, \ell + i - 1\} \setminus \{\ell + 1\}\} \\ \{0^b\} & \text{if } (u, v) = (\ell + i - 1, i) \text{ and } i > 1 \\ \{01^{b-1}\} & \text{if } (u, v) = (0, \ell + 1) \text{ and } i > 1 \\ \emptyset & \text{otherwise.} \end{cases} \end{aligned}$$

Again, it is clear from the definition of $G(i)$ that for distinct $i \neq i'$ we also have $G(i) \neq$



$\ell = 4$, $i = 3$
$m_1 = 1^b 0^{3b}$, $m_2 = 01^{b-1}0^{3b}$

Figure 6.7: A sample graph $G(i)$ for the proof of Proposition 5.

$G(i')$. Moreover, it is easy to see that for each $i \in \{1, \ldots, \ell\}$ we have $G(i) \in \mathcal{H}(\mathcal{M})$. This proves that in this case $|\mathcal{H}(\mathcal{M})| \geq \ell$ as desired. □

Finally, the ideas from the proof of Proposition 5 above can be used to give a simple non-adaptive distinguishing attack achieving advantage $\Theta(\ell q^2/2^c)$ against $\mathsf{LenCasc}_{\mathbf{0}}^{\mathbf{f}_1} \triangleright \mathbf{f}_2$, i.e., against the system that we obtain after replacing h in NI^{h} by a random compression function. We sketch this attack below, hence showing that the information-theoretic analysis in Theorem 4 is tight.

The adversary simply chooses q messages $m_1, \ldots, m_q$ of the form $m_i = x_i \| 0^{b(\ell-1)}$ for arbitrary distinct x_i's. For any $1 \leq i < j \leq q$ and any $1 \leq p \leq \ell$, we will have a collision $\mathbf{f}_2(\mathsf{LenCasc}_{\mathbf{0}}^{\mathbf{f}_1}(m_i)) = \mathbf{f}_2(\mathsf{LenCasc}_{\mathbf{0}}^{\mathbf{f}_1}(m_j))$ if the outputs after computing the inner cascade $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$ on the p-block prefixes of m_i and m_j collide (as their suffixes and lengths are identical, and thus such a collision implies that also the final values collide). The probability that for any fixed (i, j, p) this happens, conditioned on that this collision is not predetermined (i.e., either $p = 1$ or $\mathsf{Casc}_{\mathbf{0}}^{\mathbf{f}_1}$ applied to the $(p-1)$-block prefixes did not collide) is roughly 2^{-c} as long as $\ell \ll 2^{c/2}$. We can choose triples (p, i, j) in $\ell q(q-1)/2 = \Theta(\ell q^2)$ ways, and as just explained every such triple defines a possible event that leads to a collision and has probability $\approx 2^{-c}$ (and these events are disjoint as we required the collisions not to be predetermined), hence this gives the claimed $\Theta(\ell q^2/2^c)$ bound.

7 Paper 2

The Exact Security of PMAC[1]

Peter Gaži, Krzysztof Pietrzak, Michal Rybár

IST Austria

January 2017

Abstract. PMAC is a simple and parallel block-cipher mode of operation, which was introduced by Black and Rogaway at Eurocrypt 2002. If instantiated with a (pseudo)random permutation over n-bit strings, PMAC constitutes a provably secure variable input-length (pseudo)random function. For adversaries making q queries, each of length at most ℓ (in n-bit blocks), and of total length $\sigma \leq q\ell$, the original paper proves an upper bound on the distinguishing advantage of $O(\sigma^2/2^n)$, while the currently best bound is $O(q\sigma/2^n)$. In this work we show that this bound is tight by giving an attack with advantage $\Omega(q^2\ell/2^n)$.

In the PMAC construction one initially XORs a mask to every message block, where the mask for the ith block is computed as $\tau_i := \gamma_i \cdot L$, where L is a (secret) random value, and γ_i is the i-th codeword of the Gray code. Our attack applies more generally to any sequence of γ_i's which contains a large coset of a subgroup of $GF(2^n)$.

We then investigate, if the security of PMAC can be further improved by using τ_i's that are k-wise independent, for $k > 1$ (the original distribution is only 1-wise independent). We observe that the security of PMAC will not increase in general, even if the masks are chosen from a 2-wise independent distribution, and then prove that the security increases to $O(q^2/2^n)$, if the τ_i are 4-wise independent. Due to simple extension attacks, this is the best bound one can hope for, using any distribution on the masks. Whether 3-wise independence is already sufficient to get this level of security is left as an open problem.

Keywords: Message Authentication Codes, PMAC, Attack, Masks

7.1 Introduction

PMAC (for Parallelizable Message Authentication Code) is a block-cipher mode of operation, introduced by Black and Rogaway at Eurocrypt 2002 [14]. The mode, when instantiated with a block-cipher over $\{0,1\}^n$, constitutes a variable input-length pseudorandom function $\{0,1\}^* \to \{0,1\}^n$ (which is then typically used for message authentication, hence the name). PMAC is slightly less efficient than, for example, modes based on CBC MAC, but its main advantage is that unlike CBC-based MACs, it allows to process the message blocks fully in parallel.

[1] This paper appeared in the IACR Transactions on Symmetric Cryptology(ToSC) 2017, volume 1. This research was supported by the European Research Council, ERC consolidator grant (682815-TOCNeT).

The secret key of PMAC specifies two permutations π, π' over $\{0,1\}^n$, and a function $\tau : \mathbb{N} \to \{0,1\}^n$ for determining the masks. On input a message $M = m_1 \| \dots \| m_\ell, m_i \in \{0,1\}^n$, the output is computed as

$$\mathsf{PMAC}_{\pi,\pi',\tau}(M) = \pi' \left(\bigoplus_{i=1}^{\ell} \pi(m_i \oplus \tau(i)) \right) . \tag{7.1}$$

In [14], the key is just a single key $K \in \mathcal{K}$ for a block-cipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, π, π' are instantiated both with $E(K,.)$, and the mask function is defined as $\tau(i) = \gamma_i \cdot L$, where γ_i is the ith Gray codeword[2] and $L = E(K,0)$. This is a slightly idealized version of PMAC, we will discuss all the simplifications we make in greater detail in Section 7.1.3.

7.1.1 Security of PMAC in the Random Permutation Model

The security of a block-cipher mode of operation is usually analyzed assuming the underlying block-cipher under a random secret key realizes a uniformly random permutation. A bound in this model then implies security when instantiated with a block-cipher, we just have to add an extra term which bounds the advantage of distinguishing the block-cipher from a random permutation (i.e., the PRP security of the block-cipher, cf. Eq.(7.5) in this paper).

[14] proved an upper bound of $\sigma^2/2^n$ on the distinguishing advantage against PMAC for any adversary making a total of q queries, each of length at most ℓ blocks (of n bits), and a total of $\sigma \leq \ell q$ blocks. This was later improved to $q^2\ell/2^n$ by Minematsu and Matsushima at FSE'07 [49], and then to $q\sigma/2^n$ by Nandi at FSE'10 [51] (note that $q\sigma$ can be much less than $q^2\ell$, if the message lengths vary a lot).

In this work we show that this bound is tight by giving an attack with advantage $\Omega(q^2\ell/2^n)$. For this, we show that it is possible to construct q messages $M_1, \dots, M_q$ ($M_a = m_1^{(a)} \| m_2^{(a)} \| \dots \| m_\ell^{(a)}$), such that for any pair of messages (M_a, M_b),

$$\bigoplus_{i=1}^{\ell} \pi(m_i^{(a)} \oplus \tau(i)) = \bigoplus_{i=1}^{\ell} \pi(m_i^{(b)} \oplus \tau(i)) \tag{7.2}$$

for $\ell - 1$ different choices of L (where $\tau(i) = \gamma_i \cdot L$). Thus, also the PMAC tags of M_a, M_b (which additionally permutes the value in Eq. 7.2) will collide. This directly gives a distinguishing attack, and even a forgery as now $M_a \| X$ and $M_b \| X$ will collide for any string X. Moreover, the set of L's for which two messages collide will be mostly disjoint for the $\binom{q}{2}$ pairs of messages, so with q messages of length ℓ we will observe a collision with probability in the order of $q^2\ell/2^n$.

Recently, Luykx *et al.* [42] showed that one can construct a pair of messages which will collide with probability roughly $\ell/2^n$, leading to an attack with advantage $\ell/2^n$ for $q = 2$ messages. However, their attack does not generalize to q messages. In contrast, our attack obtains this high collision probability for every of the $\binom{q}{2}$ message pairs.

7.1.2 k-wise Independent Masks

Several works show that by somewhat changing the construction, one can boost the security of PMAC [65, 66, 68] even beyond the $q^2/2^n$ birthday bound. We investigate whether

[2] This encoding is chosen to allow for efficient sequential computation of the values $\gamma_1 \cdot L, \gamma_2, \cdot L, \dots$.

one can make the original construction more secure by just changing the distribution of the masks.

As a warm-up, in Section 7.4 we prove that if the masks $\tau_1, \tau_2, \ldots$ are uniform and independent, then the security indeed increases to $O(q^2/2^n)$. This is the best we can hope for under any distribution of masks: One can always query on random messages, and if a collision is found (which occurs with probability $q^2/2^n$), add the same block to both colliding messages, which will also lead to the same output.

The original distribution of masks in PMAC is only 1-wise independent, so we investigate if the security increases when using k-wise independent distributions for $k > 1$. In Section 7.6, we show that 2-wise independence in general does not increase security by constructing a 2-wise independent distribution which, for any set of messages, gives us exactly the same collision probability as the original distribution. In Section 7.5 we show that using any 4-wise independent distribution on masks[3] will boost the security to the optimal $O(q^2/2^n)$. Whether 3-wise independence is sufficient is left as an open problem.

7.1.3 Variants of the PMAC Construction

The construction that we analyze is a somewhat simplified version of the actual original PMAC as proposed in [14]. We now discuss the existing differences and the applicability of our results to other variants.

One difference is that [14] specifies a padding which allows it to take as inputs messages whose length is not a multiple of n, moreover, the last block is not permuted. Additionally, a final mask (which is fixed and independent of ℓ) is XORed to the state before the outer permutation is applied. Our attacks and security proofs can be easily adapted to take these things into account, we chose not to do so for the sake of conceptual and notational simplicity. In particular, for our attack we choose q messages for the "simplified" PMAC as in Eq. (7.1) in a way that maximises the probability of seeing a collision. XORing a fixed value to the state before applying the outer permutation does not affect this collision probability. To handle the fact that the last block is not permuted we can simply add an arbitrary dummy message block to every message. Again, this will not affect the collision probability.

Another difference is that for our security proofs we assume that the value L used for the masks is sampled uniformly at random, while in the original construction $L := \pi(0)$. This distinction does not matter as long as $\ell q \ll 2^n$ (as then whp. none of the internal queries made is 0), which is satisfied for our main security result (Lemma 15) using 4-wise independent masks, as there we must assume $\ell \leq 2^{n/2}$ anyway. For our "warm-up" proof (Lemma 14) using independent random masks we don't have to make such an assumption, so here it's not clear if the result still applies with this difference for very large ℓ. This distinction also doesn't affect the success probability of our attack, which works for any distribution on L.

Moreover, in the security proofs we also assume that the inner and outer permutations π, π' are independent, while in the original construction π and π' are the same. If one aims for security in the order of $q^2\ell/2^n$ (or more generally $\sigma\ell/2^n$), this can be handled: informally, as there are q queries to π' and $q\ell$ queries to π, we expect them to overlap only with probability $q^2\ell/2^n$, and as long as they do not overlap, we can treat them as if they

[3] For example computed as $\tau_i = \sum_{j=0}^{3} L_j \cdot i^j$ for random $L_j \in GF(2^n)$.

were independent. As we aim for $q^2/2^n$ security, it is not clear whether assuming that π and π' are independent is without loss of generality. Again, for our attack this distinction does not matter, the collision probability is the same no matter what π' is.

Let us also mention that there exists a later variant of PMAC due to Rogaway [58] called PMAC1, which for efficiency reasons deviates slightly from PMAC by using a different sequence for the γ_i values. It is not clear if our attack can be adapted to this case. Informally, we require the sequence of $\gamma_1, \ldots, \gamma_\ell$ to contain a large coset of a subgroup of $GF(2^n)$, and it's not clear if the sequence from [58] contains such a set. Let us mention that for similar reasons the attack from [42] does not apply to the [58] construction either.

Newer variations of PMAC include PMAC+ [65], PMAC with parity [66], and PMACX [68]. These introduce major modifications to the original constructions, therefore we do not discuss them in more detail. Lastly, LightMAC [43] can be considered a PMAC-like construction.

7.2 Preliminaries

Basic Definitions. For $n \in \mathbb{N}$ we define $[n] := \{1, \ldots, n\}$, and $\{0,1\}^{n*} := \bigcup_{z \in \mathbb{N}} \{0,1\}^{nz}$ denotes the set of all bitstrings whose length is a multiple of n. In a slight abuse of notation, we interchangeably view strings from $\{0,1\}^{n*}$ also as finite sequences of blocks from $\{0,1\}^n$, i.e., for $s \in \{0,1\}^{nz}$ we also write $s = (s_1, \ldots, s_z)$ for $s_i \in \{0,1\}^n$. The (bit)length of a string w is $|w|$, and if $|w|$ is a multiple of n, $|w|_n = |w|/n$ denotes the length in n bit blocks. $w^\ell := w\|w\|\ldots\|w$ denotes the ℓ-fold concatenation of w. We usually denote sets by calligraphic letters like $\mathcal{X}$. $\mathcal{F}_{b,c}$ (resp. $\mathcal{F}_{b*,c}$) denotes the set of all functions from $\{0,1\}^b$ to $\{0,1\}^c$ (resp. from $\{0,1\}^{b*}$ to $\{0,1\}^c$), $\mathcal{F}_{\mathbb{N},b}$ is the set of all functions $\mathbb{N} \to \{0,1\}^b$ and $\mathcal{P}_n$ the set of all permutations on $\{0,1\}^n$. If P is a (finite or infinite) progression, then by $P_{[\ell]}$ we denote a tuple containing the first ℓ elements of P. A *partition* of a set S is a collection of non-empty subsets A_i, such that if $A_i \neq A_j$, then $A_i \bigcap A_j = \emptyset$, and $\bigcup A_i = S$.

Multisets. We denote with $\mathsf{mult}(x, \mathcal{X})$ the multiplicity of an element x in a multiset $\mathcal{X}$. $\mathcal{X}^\downarrow$ is the subset of $\mathcal{X}$ that contains only the elements of odd multiplicity, i.e.,

$$\mathcal{X}^\downarrow = \{x \in \mathcal{X} : \mathsf{mult}(x, \mathcal{X}) \mod 2 = 1\} .$$

Groups and Cosets. For a definition of a commutative group and a discussion of the notions introduced below, see e.g. [33]. All the groups that we consider in this paper will be commutative, and we will use additive notation for groups. A *subgroup* of G is any subset H that is a group by itself. The *order* of G, denoted $|G|$ is the number of its elements. Lagrange's theorem states that if H is a subgroup of G, then $|H|$ divides $|G|$.

Let G be a group, and H its subgroup. Take $g \in G$. Then the set $g + H := \{g + h : h \in H\}$ is called a *coset of H in G*. Note that trivially any group G is a coset (of G in G), we call a coset *proper* if it is not a group. The set of different cosets of H in G forms a partition of G; and moreover, H itself appears in it as the coset $0 + H$, where 0 is the neutral element of G (and H). The size of a coset is again referred to as its *order*. Finally, the order of G is equal to the product of the order of H and the number of different cosets of H.

Random Variables and Experiments. Random variables and concrete values they can take are usually denoted by upper-case letters $X, Y, \ldots$, and lower-case letters $x, y, \ldots$ respectively.

If $\mathcal{M}$ is a distribution (respectively, a set), then we denote by $X \overset{\$}{\leftarrow} \mathcal{M}$ sampling the random variable X according to $\mathcal{M}$ (respectively, choosing it uniformly at random from $\mathcal{M}$). By X^ℓ we denote ℓ independent and identically distributed copies of a random variable X. A joint probability distribution of q random variables $(X_1, \ldots, X_q)$ is *k-wise independent*, if its restriction to any k coordinates is uniform over its domain, e.g., if all X_i have domain $\{0,1\}^n$

$$\forall i_1, \ldots, i_k \ , \ 1 \le i_1 < \cdots < i_k \le q \ ; \ \forall x_1, \ldots, x_k \in \{0,1\}^n \ :$$
$$\mathsf{Pr}_{(X_1,\ldots,X_q)}\left((X_{i_1}, \ldots, X_{i_k}) = (x_1, \ldots, x_k)\right) = \left(2^{-n}\right)^k \ .$$

More generally, let $\mathcal{M}_n$ be a probability distribution over $\mathcal{F}_{\mathbb{N},n}$. In this case, we call $\mathcal{M}_n$ *k-wise independent*, if any k outputs of $f(\cdot)$ sampled from $\mathcal{M}_n$ are independent. Formally, $\mathcal{M}_n$ is k-wise independent, if:

$$\forall i_1, \ldots, i_k \ , \ 1 \le i_1 < \cdots < i_k \ ; \ \forall x_1, \ldots, x_k \in \{0,1\}^n \ :$$
$$\underset{f \overset{\$}{\leftarrow} \mathcal{M}_n}{\mathsf{Pr}} \Big(\big(f(i_1), \ldots, f(i_k)\big) = (x_1, \ldots, x_k) \Big) = \left(2^{-n}\right)^k \ .$$

Adversaries. In this work an adversary is a probabilistic (polynomial time or computationally unbounded) algorithm, sometimes with access to an oracle $\mathcal{O}(\cdot)$. We use sans-serif letters for adversaries, e.g., $\mathsf{A}^{\mathcal{O}(\cdot)}$, and will only consider "distinguishers", which are adversaries, whose final output is just one bit.

Pseudorandom functions and permutations. We call a function $f : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ *keyed*, where the first part of the input is referred to as the key (and $\mathcal{K}$ being called the *keyspace* of f). We often write $f_k(\cdot)$ instead of $f(k, \cdot)$. Given a variable input-length keyed function $f \colon \mathcal{K} \times \{0,1\}^{n*} \to \{0,1\}^n$, the PRF-advantage of an adversary A against f is defined as

$$\mathbf{Adv}_f^{\mathrm{prf}}(\mathsf{A}) := \mathsf{Pr}[K \overset{\$}{\leftarrow} \mathcal{K} \ : \ \mathsf{A}^{f_K(\cdot)} = 1] - \mathsf{Pr}[\mathsf{R} \overset{\$}{\leftarrow} \mathcal{F}_{n*,n} \ : \ \mathsf{A}^{\mathsf{R}(\cdot)} = 1] \ .$$

We also define

$$\mathbf{Adv}_f^{\mathrm{prf}}(q, \ell, t) := \max_{\mathsf{A}} \mathbf{Adv}_f^{\mathrm{prf}}(\mathsf{A})$$

where the maximum goes over all adversaries that run in time at most t, and ask at most q queries, each of length at most ℓ (in n-bit blocks). If we consider computationally unbounded adversaries, we drop the last argument, i.e., $\mathbf{Adv}_f^{\mathrm{prf}}(q, \ell) := \mathbf{Adv}_f^{\mathrm{prf}}(q, \ell, \infty)$.

Pseudorandom permutations (PRPs), and their security notions are defined analogously. Given a keyed permutation (i.e., a block-cipher) $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, the PRP-advantage of an adversary A against E is defined as

$$\mathbf{Adv}_E^{\mathrm{prp}}(\mathsf{A}) := \mathsf{Pr}[K \overset{\$}{\leftarrow} \mathcal{K} \ : \ \mathsf{A}^{E_K(\cdot)} = 1] - \mathsf{Pr}[\mathsf{P} \overset{\$}{\leftarrow} \mathcal{P}_n \ : \ \mathsf{A}^{\mathsf{P}(\cdot)} = 1] \ .$$

and

$$\mathbf{Adv}_E^{\mathrm{prp}}(q, t) := \max_{\mathsf{A}} \mathbf{Adv}_E^{\mathrm{prp}}(\mathsf{A})$$

where the maximum goes over all adversaries that run in time at most t and ask at most q queries.

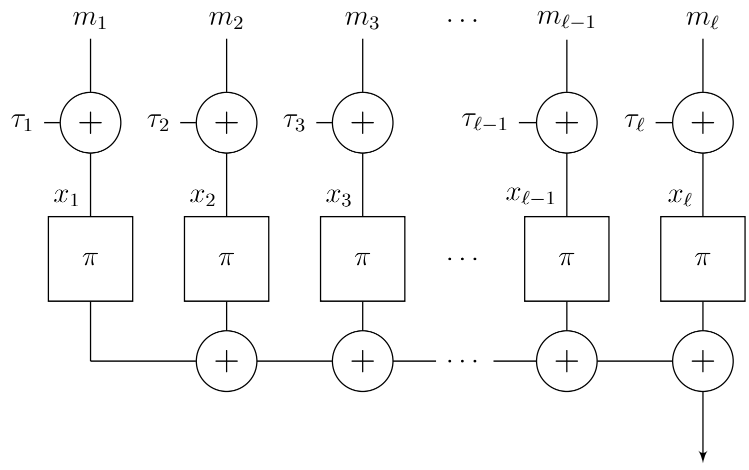


Figure 7.1: The evaluation of $\mathsf{sPMAC}(\pi, \tau, m_1\| \dots \|m_\ell)$, where $\tau_i = \tau(i)$.

Collision security. For a keyed function $f\colon \mathcal{K} \times \{0,1\}^{n*} \to \{0,1\}$, we define

$$\mathbf{Adv}_f^{\mathrm{col}}(q, \ell) := \max_{M_1,\dots,M_q} \mathsf{Pr}_{K \leftarrow \mathcal{K}} \left[\exists\, i \neq j \ :\ f_K(M_i) = f_K(M_j)\right] \ ,$$

where the maximum goes over all q tuples of distinct messages of length at most ℓ blocks.

7.3 PMAC and Simplified PMAC

We define the simplified PMAC, $\mathsf{sPMAC}\colon \mathcal{P}_n \times \mathcal{F}_{\mathbb{N},n} \times \{0,1\}^{n*} \to \{0,1\}^n$ as

$$\mathsf{sPMAC}(\pi, \tau, m_1\| \dots \|m_\ell) := \bigoplus_{i=1}^{\ell} \pi(m_i \oplus \tau(i)) \ .$$

$\mathsf{PMAC} : \mathcal{P}_n \times \mathcal{P}_n \times \mathcal{F}_{\mathbb{N},n} \times \{0,1\}^{n*} \to \{0,1\}^n$ is derived from sPMAC by additionally encrypting the final output using an independent permutation π':

$$\mathsf{PMAC}(\pi, \pi', \tau, M) = \pi'(\mathsf{sPMAC}(\pi, \tau, M))$$

To save on notation, we will sometimes write τ_i instead $\tau(i)$, and e.g., $\mathsf{PMAC}_{\pi,\pi',\tau}(M)$ instead of $\mathsf{PMAC}(\pi, \pi', \tau, M)$, or, if π, π', τ are clear from the context, simply $\mathsf{PMAC}(M)$.

The first three (two) arguments of PMAC (sPMAC) are the key; consider distributions Π, Π' over $\mathcal{P}_n$, and $\mathtt{T}_n$ over $\mathcal{F}_{\mathbb{N},n}$, then $\mathsf{PMAC}_{\Pi,\Pi',\mathtt{T}_n}(.)$ denotes a keyed function, where the key is sampled according to $(\pi, \pi', \tau) \leftarrow \Pi \times \Pi' \times \mathtt{T}_n$, and then defines the function $\mathsf{PMAC}_{\pi,\pi',\tau}(.)$. If Π is instantiated by a block-cipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, we think of it as the uniform distribution over the multiset of functions $\{E(k, \cdot) \ :\ k \in \mathcal{K}\}$.

For an input message $M = m_1\| \dots \|m_\ell$, it will be convenient to define the following variables

$$\begin{aligned} x_i &:= m_i \oplus \tau_i,\ \forall i \quad ; \quad \mathcal{X} := (x_1, \dots, x_\ell) \\ y_i &:= \pi(x_i),\ \forall i \quad ; \quad \mathcal{Y} := (y_1, \dots, y_\ell) \end{aligned} \tag{7.3}$$

We often consider pairs of messages $M = m_1 \| \ldots \| m_s, M' = m'_1 \| \ldots \| m'_{s'}$, and so $\mathcal{X}^*$ denotes the multiset

$$x_i := m_i \oplus \tau_i \ , \ x'_i := m'_i \oplus \tau_i, \ \forall i \qquad ; \qquad \mathcal{X}^* := (x_1, \ldots, x_s, x'_1, \ldots, x'_{s'}) \tag{7.4}$$

We start by reducing the PRP-security of PMAC with a block-cipher E to the collision security of sPMAC with a random permutation. The argument is fairly standard and allows us to perform the rest of our analysis in the information-theoretic setting.

Lemma 12 (PRF security of PMAC from collision security of sPMAC)**.** *For a block-cipher $E : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$, and for any distribution $\mathtt{T}_n$ over $\mathcal{F}_{\mathbb{N},n}$, we have*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{E,E,\mathtt{T}_n}}(q, \ell, t) \leq 2 \cdot \mathbf{Adv}^{\mathrm{prp}}_{E}(\ell q, t') + \mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(q, \ell) + \frac{q^2}{2^n} \ ,$$

where $t' \leq t + O(\ell q)$.

Proof. We first replace the block-cipher E with uniformly random permutations, by a straightforward reduction:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{E,E,\mathtt{T}_n}}(q, \ell, t) \leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{\mathcal{P}_n \mathcal{P}_n,\mathtt{T}_n}}(q, \ell, t) + 2 \cdot \mathbf{Adv}^{\mathrm{prp}}_{E}(\ell q, t') \ . \tag{7.5}$$

We can now consider computationally unbounded distinguishers (first step below), and replace the outer permutation by a uniformly random function, using the PRF/PRP switching lemma [12] in the second step:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{\mathcal{P}_n,\mathcal{P}_n,\mathtt{T}_n}}(q, \ell, t) \leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{\mathcal{P}_n,\mathcal{P}_n,\mathtt{T}_n}}(q, \ell) \leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{\mathcal{P}_n,\mathcal{F}_{n,n},\mathtt{T}_n}}(q, \ell) + \frac{q^2}{2^n} \tag{7.6}$$

Finally, we claim that distinguishing $\mathsf{PMAC}_{\mathcal{P}_n,\mathcal{F}_{n,n},\mathtt{T}_n}$ from a random function is upper bounded by the collision security of $\mathsf{sPMAC}_{\pi,\tau}$, i.e.,

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{\mathcal{P}_n,\mathcal{F}_{n,n},\mathtt{T}_n}}(q, \ell) \leq \mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(q, \ell) \tag{7.7}$$

The statement of the Lemma follows from Eq.(7.5)-Eq.(7.7). It remains to prove Eq.(7.7). As the outer function $f \leftarrow \mathcal{F}_{n,n}$ is uniformly random, the output of $\mathsf{PMAC}_{\pi,f,\tau}(.) \equiv f(\mathsf{sPMAC}_{\pi,\tau}(.))$ is uniformly random, conditioned on not having any collisions on the inner function $\mathsf{sPMAC}_{\pi,\tau}(.)$. By Theorem 1.(i) from [46], this implies that distinguishing $\mathsf{PMAC}_{\pi,f,\tau}(.)$ from a random function is at least as hard as provoking a collision on $\mathsf{sPMAC}_{\pi,\tau}(.)$, and further by Theorem 2 from [46], adaptivity does not help in provoking this condition. This concludes the proof of Eq.(7.7). □

A *self-cancellation* for a message M (denoted $\mathsf{seCan}(M)$) occurs, if for its corresponding $\mathcal{X}$, we have $\mathcal{X}^{\downarrow} = \emptyset$. A *cross-cancellation* for two messages M, M' (denoted $\mathsf{crCan}(M, M')$) occurs, if for their corresponding $\mathcal{X}^{*\downarrow}$, we have $\mathcal{X}^{*\downarrow} = \emptyset$. A *PMAC-collision* for two messages M, M' (denoted $\mathsf{pCol}(M, M')$) occurs, if $\mathsf{PMAC}(M) = \mathsf{PMAC}(M')$. We define sPMAC-*collision* ($\mathsf{spCol}(M, M')$) analogously. Note that $\mathsf{crCan}(M, M')$ implies $\mathsf{spCol}(M, M')$, and $\mathsf{spCol}(M, M')$ implies $\mathsf{pCol}(M, M')$.

For a given n, ℓ, and a distribution $\mathtt{T}_n$, we define the following quantity with the x_i's as defined in Eq.(7.3):

$$\theta(\ell, n, \mathtt{T}_n) = \max_{\substack{M \neq M' \\ |M|_n, |M'|_n \leq \ell}} \Pr_{\tau \xleftarrow{\$} \mathtt{T}_n} \left[\{x_1, x_2, \ldots, x_{|M|_n}, x'_1, x'_2, \ldots, x'_{|M'|_n}\}^{\downarrow} = \emptyset \right] \ . \tag{7.8}$$

The quantity $\theta(\ell, n, \mathtt{T}_n)$ bounds the maximum probability over all pairs of distinct messages M, M' of maximum length ℓ that their reduced set $\mathcal{X}^{*\downarrow}$ is empty, and hence a cross-cancellation occurs. This probability is taken over the sampling of the mask according to the distribution $\mathtt{T}_n$.

The following lemma states that a cross-cancellation is indeed the dominant reason for an sPMAC-collision to occur.

Lemma 13. *For any* $n, \mathtt{T}_n$*, and* $\ell \leq 2^{n-2}$

$$\mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(q,\ell) \leq \theta(\ell, n, \mathtt{T}_n) \cdot q^2 + \frac{q^2}{2^{n-1}} \; .$$

Proof. By taking a union bound over all q messages, we can upper bound the probability of a collision amongst the q messages by the probability of any pair colliding as:

$$\mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(q,\ell) \leq \mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(2,\ell) \cdot \binom{q}{2} \leq \mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(2,\ell) \cdot q^2 \; .$$

We upper bound $\mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(2,\ell)$ by showing that for any $M \neq M', |M|_n, |M'|_n \leq \ell$, we have:

$$\begin{aligned}
&\mathsf{Pr}_{(\pi,\tau)\leftarrow\mathcal{P}_n\times\mathtt{T}_n}[\mathsf{sPMAC}_{\pi,\tau}(M) = \mathsf{sPMAC}_{\pi,\tau}(M')] \\
&\qquad = \mathsf{Pr}_{(\pi,\tau)\leftarrow\mathcal{P}_n\times\mathtt{T}_n}[\mathsf{sPMAC}_{\pi,\tau}(M) = \mathsf{sPMAC}_{\pi,\tau}(M') \wedge \mathsf{crCan}(M,M')] && (7.9)\\
&\qquad + \mathsf{Pr}_{(\pi,\tau)\leftarrow\mathcal{P}_n\times\mathtt{T}_n}[\mathsf{sPMAC}_{\pi,\tau}(M) = \mathsf{sPMAC}_{\pi,\tau}(M') \wedge \overline{\mathsf{crCan}(M,M')}] && (7.10)\\
&\qquad \leq \theta(\ell, n, \mathtt{T}_n) + \frac{1}{2^n - 2\ell} \\
&\qquad \leq \theta(\ell, n, \mathtt{T}_n) + \frac{1}{2^{n-1}} && (7.11)
\end{aligned}$$

Note that this proves the statement of the Lemma. In eq. (7.11), we have used $\ell \leq 2^{n-2}$. The term (7.9) can be upper bounded as (using that for any events E_0, E_1,$\mathsf{Pr}[E_0 \wedge E_1] \leq \mathsf{Pr}[E_0]$)

$$(7.9) \leq \mathsf{Pr}_{(\pi,\tau)\leftarrow\mathcal{P}_n\times\mathtt{T}_n}[\mathsf{crCan}(M,M')] \leq \theta(\ell, n, \mathtt{T}_n) \; ,$$

where the 2nd step follows by definition.

It remains to upper bound the term (7.10) by $1/(2^n - 2\ell)$. We first upper bound (7.10) by fixing τ to the "worst case", and condition on crCan (using $\mathsf{Pr}[E_0 \wedge E_1] \leq \mathsf{Pr}[E_0|E_1]$)

$$(7.10) \leq \max_{\tau} \mathsf{Pr}_{\pi\leftarrow\mathcal{P}_n}[\mathsf{sPMAC}_{\pi,\tau}(M) = \mathsf{sPMAC}_{\pi,\tau}(M') \mid \overline{\mathsf{crCan}(M,M')}]$$

As $\overline{\mathsf{crCan}(M,M')}$, the set $\mathcal{X}^{*\downarrow} = \{a_1, \ldots, a_s\}$ is non-empty and $s \leq 2\ell$. A necessary (albeit not sufficient) condition to have a collision is that $\bigoplus_{i=1}^{s} \pi(a_i) = 0$. We claim that

$$\mathsf{Pr}_{\pi\leftarrow\mathcal{P}_n}\left[\bigoplus_{i=1}^{s} \pi(a_i) = 0\right] = \mathsf{Pr}_{\pi\leftarrow\mathcal{P}_n}\left[\bigoplus_{i=1}^{s-1} \pi(a_i) = \pi(a_s)\right] \leq \frac{1}{2^n - s + 1}$$

The first equality follows as $A \oplus B = 0$, if and only if $A = B$. To see the second step, assume the output of the random π is defined in a lazy way (sampling a random image without repetition for every fresh input), starting with inputs $a_1, \ldots, a_{s-1}$. Once these have been defined, we know that $\pi(a_s)$ will be uniform over a set of size $2^n - s + 1$, but at most one value, namely $\pi(a_s) = \bigoplus_{i=1}^{s-1} \pi(a_i)$, will satisfy the required condition. □

7.4 Independent Random Masks

In this section, as a warm-up, we look at the setting where the masks are chosen independently and uniformly at random.

Lemma 14. *For any $n, \ell \in \mathbb{N}$*

$$\theta(n, \ell, \mathcal{F}_{\mathbb{N},n}) \leq \frac{2}{2^n} .$$

Before we prove the lemma, we note that this upper bound is tight: consider the messages $M = 0^n \| 0^n$ and $M' = 1^n \| 1^n$, then for any choice of τ_1, we'll have $\mathcal{X}^{*\downarrow} = \{\tau_1, \tau_2, 1^n \oplus \tau_1, 1^n \oplus \tau_2\} = \emptyset$, if either $\tau_2 = \tau_1$, or $\tau_2 = \tau_1 \oplus 1^n$. As the τ_i are uniform, $\mathsf{Pr}[(\tau_1 = \tau_2) \vee \tau_1 = \tau_2 \oplus 1^n] = 2/2^n$.

Proof. Recall that $\theta(n, \ell, \mathcal{F}_{\mathbb{N},n})$ is the probability, maximized over all $M \neq M'$ of length $|M|_n, |M'|_n \leq \ell$, that $\mathcal{X}^{*\downarrow} = \emptyset$. Let $M = m_1 \| \ldots \| m_s$, $M' = m'_1 \| \ldots \| m'_{s'}$ denote the messages maximising this probability.

If $s = s'$, i.e., the messages are of same length, then let i be the smallest index where $m_i \neq m'_i$ (this i exists as $M \neq M'$). Assume $\tau \leftarrow \mathcal{F}_{\mathbb{N},n}$ is sampled on all inputs, except i. Consider the multiset $\mathcal{Z}_i = \mathcal{X}^* - (x_i, x'_i)$. Now, we claim $\mathcal{X}^{*\downarrow}$ will be empty, if and only if two conditions are satisfied. Firstly, $\mathcal{Z}_i^{\downarrow}$ must contain exactly two elements, let's call them $\{a, b\}$. If this is not the case, then $\mathcal{X}^{*\downarrow}$ will not be empty with probability 1.[4]

If it has exactly two elements $\{a, b\}$, then secondly, $\tau(i)$ must be chosen such that $\{m_i \oplus \tau(i), m'_i \oplus \tau(i)\} = \{x_i, x'_i\} = \{a, b\}$. There are at most two possible values for $\tau(i)$, which satisfy this condition (two, not just one, as the sets are not ordered). As $\tau(i)$ is uniform, the probability it hits one of those two values is $2/2^n$.

Now let us consider the case where $s \neq s'$ (without loss of generality, $s > s'$). We can't use the above argument here as an index i with $m_i \neq m'_i$ will not exist if M is a prefix of M'. In this case we assume $\tau(i)$ is given to us on all inputs except $s + 1$. Following a similar argument as with $\mathcal{Z}_i^{\downarrow}$ above, there will be at most one value for $\tau(s+1)$ that will cause $\mathcal{X}^{*\downarrow}$ to be empty, which upper bounds the probability of $\mathcal{X}^{*\downarrow} = \emptyset$ to $1/2^n$.

□

Lemma 14, in combination with Lemmas 12 and 13, directly give us the following statement.

Theorem 5 (PMAC security with uniform masks)**.** *For any q, t, n, ℓ, where $\ell \leq 2^{n-2}$, and block-cipher E with block-size n, we have*

$$\mathbf{Adv}^{\text{prf}}_{\mathsf{PMAC}_{E,E,\mathcal{F}_{\mathbb{N},n}}}(q, \ell, t) \leq \frac{5q^2}{2^n} + 2 \cdot \mathbf{Adv}^{\text{prp}}_{E}(\ell q, t') ,$$

where $t' \leq t + O(\ell q)$.

[4] Intuitively, we need $\{a, b\}$ and $\{x_i, x'_i\}$ to cancel each other out for $\mathcal{X}^{*\downarrow} = \emptyset$. If $\mathcal{Z}_i^{\downarrow}$ had more, or less than precisely two elements, this would not be possible.

7.5 4-wise Independent Masks

In this section we investigate the security of PMAC if the mask distribution $\mathtt{T}_n$ is 4-wise independent. By the following lemma this assumption is sufficient to prove a bound of order $q^2/2^n$ (i.e., independent of the message length ℓ) on the PRF-security of PMAC assuming an exponential upper bound on ℓ.

Lemma 15. *For any $n, \ell \in \mathbb{N}$, where $\ell \leq 2^{n/2}$, and any 4-wise independent distribution $\mathtt{T}_n$, we have*

$$\theta(\ell, n, \mathtt{T}_n) \leq \frac{4}{2^n} . \tag{7.12}$$

Proof. Let $M = m_1 \| \ldots \| m_s, M' = m'_1 \| \ldots \| m'_{s'}; s, s' \leq \ell$ be messages maximizing the probability in the definition of $\theta(\ell, n, \mathtt{T}_n)$ (cf. Eq.(7.8)) for the 4-wise independent mask distribution $\mathtt{T}_n$.

We start with the case where $s = s'$. Let $I = \{i : m_i \neq m'_i\}$ be the indices of message blocks where the two messages differ, and $\mathcal{X}^*_I \subseteq \mathcal{X}^* = \{x_1, \ldots, x_s, x'_1, \ldots, x'_s\}$ the multiset containing only x_i, x'_i for $i \in I$. Note that $\mathcal{X}^{*\downarrow} = \mathcal{X}^{*\downarrow}_I$, since $m_i = m'_i$ implies $x_i = x'_i$ and for any multiset $\mathcal{S}$ and any x such that $\mathsf{mult}(\mathcal{S}, x) \geq 2$, we have $\mathcal{S}^{\downarrow} = (\mathcal{S} \setminus \{x, x\})^{\downarrow}$. In order to bound the probability that $\mathcal{X}^{*\downarrow} = \emptyset$, it suffices to bound the probability that $\mathcal{X}^{*\downarrow}_I = \emptyset$, let us denote this event with E_{col}.

If E_{col} holds, then we can find a complete matching (i.e., a subgraph where every vertex has degree exactly 1) in a graph, whose vertices are the elements of $\mathcal{X}^*_I$ and two vertices are connected by an edge, if and only if they have the same value. Note that if E_{col} holds, then every value appears with even multiplicity, so this graph consists of cliques of even size.

We will define a set of events $\{E_{\alpha,\beta} : (\alpha, \beta) \in (I \times \{0,1\})^2\}$ and prove that if $|I| \geq 4$ (we will discuss the cases where $|I| < 4$, and $s \neq s'$ at the end), then:

i. For any α, β, $\mathsf{Pr}[E_{\alpha,\beta}] \leq 2^{-2n}$.

ii. E_{col} implies that for some α, β the event $E_{\alpha,\beta}$ holds.

The above two points then imply $\mathsf{Pr}[E_{col}] \leq \sum_{\alpha,\beta} \mathsf{Pr}[E_{\alpha,\beta}] \leq 2^2|I|^2/2^{2n} \leq 4\ell^2/2^{2n}$, which is upper bounded by $4/2^n$ if $\ell \leq 2^{n/2}$, as claimed in the statement of the lemma.

It will be convenient to define the index of a message as $\alpha = (\alpha_i, \alpha_b) \in I \times \{0,1\}$, where $m_\alpha = m_{\alpha_i}$ if $\alpha_b = 0$ and $m_\alpha = m'_{\alpha_i}$ if $\alpha_b = 1$ (similarly for x_α), so the part α_i identifies the block number, and the bit α_b indicates whether we consider M or M'.

Let $I = \{i_1, i_2, \ldots\}$ and $\gamma = (\gamma_i, \gamma_b) = (i_1, 0)$, now the event $E_{\alpha,\beta}$ is defined as follows. Let

$$\delta = (\delta_i, \delta_b) = (\min\{I \setminus \{\gamma_i, \alpha_i, \beta_i\}\}, 0) .$$

Note that above $\min\{I \setminus \{\gamma_i, \alpha_i, \beta_i\}$ is non-empty, as $|I| \geq 4$. If $\gamma_i = \alpha_i$, or $\alpha = \beta$, the event $E_{\alpha,\beta}$ is defined to never hold, so from now on we assume this is not the case. Then $E_{\alpha,\beta}$ is defined as

$$E_{\alpha,\beta} \iff (x_\gamma = x_\alpha) \wedge (x_\delta = x_\beta) .$$

We first prove that

$$\begin{aligned}\mathsf{Pr}_{\tau\leftarrow\mathtt{T}_n}[E_{\alpha,\beta}] &= \mathsf{Pr}_{\tau\leftarrow\mathtt{T}_n}[(x_\gamma = x_\alpha)\wedge(x_\delta = x_\beta)]\\ &= \mathsf{Pr}_{\tau\leftarrow\mathtt{T}_n}[x_\gamma = x_\alpha]\mathsf{Pr}_{\tau\leftarrow\mathtt{T}_n}[x_\delta = x_\beta|x_\gamma = x_\alpha]\\ &= 2^{-n}\cdot 2^{-n}\ .\end{aligned}$$

To see the last step above, note that

$$\mathsf{Pr}_{\tau\leftarrow\mathtt{T}_n}[x_\gamma = x_\alpha] = \mathsf{Pr}_{\tau\leftarrow\mathtt{T}_n}[m_\gamma\oplus\tau_{\gamma_i} = m_\alpha\oplus\tau_{\alpha_i}] = 2^{-n}$$

holds as $\tau_{\gamma_i}, \tau_{\alpha_i}$, coming from a 4-wise independent distribution, are uniformly random and independent (recall we assume $\alpha_i \neq \gamma_i$). To show

$$\mathsf{Pr}_{\tau\leftarrow\mathtt{T}_n}[x_\delta = x_\beta|x_\gamma = x_\alpha] = \mathsf{Pr}_{\tau\leftarrow\mathtt{T}_n}[m_\delta\oplus\tau_{\delta_i} = m_\beta\oplus\tau_{\beta_i}|m_\gamma\oplus\tau_{\gamma_i} = m_\alpha\oplus\tau_{\alpha_i}] = 2^{-n} \quad (7.13)$$

we note that, as the τ_i are 4-wise independent, and $\delta_i \notin \{\alpha_i, \beta_i, \gamma_i\}$, the τ_{δ_i} is uniformly random even given all the other masks $\tau_{\alpha_i}, \tau_{\beta_i}, \tau_{\gamma_i}$. This concludes the proof of the condition (i), establishing that $\mathsf{Pr}[E_{\alpha,\beta}] \leq 2^{-2n}$.

It remains to show condition (ii), claiming that E_{col} implies that for some α, β the event $E_{\alpha,\beta}$ holds. For this we simply note that if E_{col} holds, then x_γ (with γ as defined above) must collide with at least some value x_α, and then the value x_δ (with δ as defined above) must collide with some x_β, thus $E_{\alpha,\beta}$ holds.

We have so far assumed that $|I| \geq 4$ and $s = s'$. If $|I| < 4$ (but we still assume $s = s'$), then there are at most $2(|I|-1) = 4$ possible values x_γ can collide with, this probability is easily upper bonded by $4/2^n$ (2-wise independence of the τ_i is sufficient here). As x_γ colliding with another value x_α (where $\alpha_i \in I$) is a necessary condition for E_{col} to hold, the same upper bound holds of E_{col}.

We now shortly describe how to adapt the proof if the messages have different lengths, say $s > s'$. Let I again denote the set of indices $i \in \{1, \ldots, s'\}$, such that $m_i \neq m'_i$.

If $2|I| + (s' - s) \leq 6$ then we use basically the same argument as for the $|I| < 4$ case above; To have the event E_{col} the value x_γ (where $\gamma = (\min\{I\}, 0)$, or if $|I| = 0$, $\gamma = (s'+1, 0)$) must collide with some x_α, and as there are at most 4 possibilities for α, this probability is at most $4/2^n$. If $2|I| + (s' - s) > 6$ then we have at least 4 indices (namely I and $s'+1, \ldots, s$) which correspond to x's that must collide, and for this one can use a slight generalisation of the argument for $|I| \geq 4$ from above. □

Again, combining Lemma 15 with Lemmas 12 and 13 give us the following statement.

Theorem 6 (PMAC security with 4-wise independent masks)**.** *For any q, t, n and $\ell \leq 2^{n/2}$, any block-cipher E with block-size n, and any 4-wise independent distribution $\mathtt{T}_n$ over $\mathcal{F}_{\mathbb{N},n}$, we have*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{E,E,\mathtt{T}_n}}(q, \ell, t) \leq \frac{7q^2}{2^n} + 2\cdot\mathbf{Adv}^{\mathrm{prp}}_{E}(\ell q, t')\ ,$$

where $t' \leq t + O(\ell q)$.

7.6 2-wise Independent Masks

In Section 7.5, we showed that the security of PMAC with 4-wise independent masks is $q^2/2^n$. On the other hand, in Section 7.7 we will show that when using the original distribution on masks from [14], which is only 1-wise independent, the security is just $\ell q^2/2^n$. This leaves open the question, whether we can get $q^2/2^n$ security already using any 2-wise or 3-wise independent distribution on masks. Below, we show that using a 2-wise independent distribution will in general not improve security: We slightly change the original distribution to make it 2-wise independent, and observe that this does not change the collision probability of sPMAC, and thus also attacker's distinguishing advantage of PMAC in the ideal permutation model, at all. Whether 3-wise independence is sufficient is left as an open problem.

Recall that in [14] the masks are computed by means of a function chosen at random from the following family

$$\{i \to a \cdot p_i \mid a \in GF(2^n)\} ,$$

where p_i is the i-th Gray codeword. For the following argument $P = (p_1, p_2 \dots, p_{2^n})$ can be any progression without repetitions. Let $\mathtt{T}_n$ denote this distribution, and note that it is 1-wise, but not 2-wise, independent. Let $\mathtt{T}_n^+$ denote the uniform distribution over

$$\{i \to a \cdot p_i \oplus b \mid a, b \in GF(2^n)\} ,$$

which is 2-wise independent.

By the following lemma, the collision security of sPMAC is exactly the same for $\mathtt{T}_n$ and $\mathtt{T}_n^+$, thus also the security of PMAC implied by Lemma 12 will be the same for both distributions.

Lemma 16. *Let $\mathtt{T}_n$ and $\mathtt{T}_n^+$ be distributions as defined above. Then, we have*

$$\mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n}}(q,\ell) = \mathbf{Adv}^{\mathrm{col}}_{\mathsf{sPMAC}_{\mathcal{P}_n,\mathtt{T}_n^+}}(q,\ell) \ .$$

Proof. Consider any messages M, M' and $\mathcal{X}^* = (x_1, \dots, x_{|M|_n}, x'_1, \dots, x'_{|M'|_n})$ where $x_i = m_i \oplus a \cdot p_i \oplus b, x'_i = m'_i \oplus a \cdot p_i \oplus b$ for random a, b, i.e., according to mask distribution $\mathtt{T}_n^+$. To prove the lemma, it is sufficient to observe that if $\mathcal{X}^{*\downarrow} = \emptyset$, then we will still have $\mathcal{X}^{*\downarrow} = \emptyset$, even if we replace b with any other element of the field, in particular, we can assume $b = 0$, in which case we get mask distribution $\mathtt{T}_n$. □

7.7 1-wise Independent Masks: PMAC with a Gray Code

In this section we analyse the PRF-security of PMAC with a one-wise independent mask distribution.

The Gray Code. The original PMAC construction uses a mask distribution based on a Gray code, which is an example of a one-wise independent distribution. A Gray code is an ordering $\gamma^\ell = \gamma_0^\ell \gamma_1^\ell \dots \gamma_{2^\ell - 1}^\ell$ of $\{0,1\}^\ell$, for any $\ell \geq 1$, such that successive points differ in precisely one bit. The canonical Gray code from [14] is defined as follows:

$$
\begin{aligned}
\gamma^1 &= (\gamma_0^1, \gamma_1^1) := (0,1)\\
\gamma^2 &= (\gamma_0^2, \gamma_1^2, \gamma_2^2, \gamma_3^2) := (00, 01, 11, 10)\\
&\ \vdots\\
\gamma^{\ell+1} &= (0\gamma_0^\ell, 0\gamma_1^\ell, \cdots, 0\gamma_{2^\ell-2}^\ell, 0\gamma_{2^\ell-1}^\ell, 1\gamma_{2^\ell-1}^\ell, 1\gamma_{2^\ell-2}^\ell, \cdots, 1\gamma_1^\ell, 1\gamma_0^\ell)
\end{aligned}
$$

In PMAC the sequence $\tau_1, \tau_2, \dots$ of masks is defined as $\tau_i := \gamma_i^n \cdot L$ for a pseudorandom $L = E_K(0)$. Let us stress that the first mask is τ_1, so the first codeword $\gamma_0^n = 0^n$ is omitted. This fact makes our attack somewhat more complicated, as the lack of the zero element in the progression $\gamma_1^n, \gamma_2^n, \dots$ will force us to argue over cosets of subgroups, instead of subgroups directly.

The [42] Attack. [42] show an attack on PMAC using two messages of length ℓ (for ℓ being any power of 2) with advantage roughly $\ell/2^n$. This attack exploits the fact that the first 2^w codewords of the *canonical* Gray code form a subgroup of the additive group of the finite field $GF(2^n)$. Hence, this two-query attack improves linearly with the increasing message length ℓ.

However, it is unclear whether this length-dependent attack can be generalized to a larger number of queries q. This is because the two attack queries are derived from the Gray code codewords being used, and are fully determined by them. Therefore, having more available message queries does not increase the success probability of the attack. Moreover, the set of L values that cause the two messages to collide on PMAC output is also predetermined by these codewords. Hence, there is a simple countermeasure against the attack: the user could simply avoid these "weak" keys.

7.7.1 Our Attack on PMAC

In this section we present an attack which scales with q, achieving success probability roughly $\ell q^2/2^n$ against PMAC. Moreover, this attack is randomized, so no "weak" keys exist, therefore a countermeasure against the [42] attack as mentioned above no longer applies.

Our attack can be mounted against PMAC using a similar class of 1-wise independent mask distributions as the attack in [42]. Namely, we assume that the masks are derived as $\tau_i := p_i \cdot R$ for some progression $P = (p_1, \dots, p_{2^n})$, where every $p_i \in \{0,1\}^n$, and a value $R \xleftarrow{\$} \{0,1\}^n$, which we model as sampled uniformly at random.[5] We assume that all elements of P are distinct (any Gray code satisfies this property by definition). Our attack differs from the one in [42] in the message construction, and the type of collisions that it is aiming for. While in [42] the authors construct a pair of messages M, M', such that $\mathsf{seCan}(M)$ and $\mathsf{seCan}(M')$ occur with probability $\ell/2^n$ (over the choice of R), we choose q messages $M_1, \dots, M_q$, such that for every pair M_i, M_j of them, $\mathsf{crCan}(M_i, M_j)$ occurs with probability $\ell/2^n$.

[5] Note that this is not completely true for the value L described above, but we can afford this imprecision when modelling an *attack*, as it obviously does not significantly affect its performance.

Algorithm 2: Attacker $\mathsf{A}^{\mathcal{O}(\cdot)}_{\ell,q,n}$ against PMAC, where $P = (p_1, \dots, p_{2^n-1})$

```
$I_S$ := indices in $P_{[\ell]}$ of a coset $S \subseteq P_{[\ell]}$ of a subgroup $H$ in an additive
    group $G \subseteq GF(2^n)$
$\ell_S := |S|$
fix arbitrary $e \in S$ (if $S$ is a group, set $e := 0$)
$I'_S$ := indices in $P_{[\ell]}$ of $S \setminus \{e\}$
$\mathcal{U}_0 := \emptyset$
for $a := 1 \dots q$ do
    repeat
        $\hat{m}^{(a)} \xleftarrow{\$} \{0,1\}^n$
    until $\left|\left\{\frac{\hat{m}^{(a)} \oplus \hat{m}^{(b)}}{e \oplus p_i} \,:\, b \in [a-1], i \in I'_S\right\} \cap \mathcal{U}_{a-1}\right| \leq \frac{2(a-1)^3(\ell_S-1)^2}{2^n}$
    $\mathcal{U}_a := \mathcal{U}_{a-1} \cup \left\{\frac{\hat{m}^{(a)} \oplus \hat{m}^{(b)}}{e \oplus p_i} \,:\, b \in [a-1], i \in I'_S\right\}$
    $M_a := \emptyset$
for $i := 1 \dots \ell$ do
    for $a := 1 \dots q$ do
        if $i \in I_S$ then
            $M_a := M_a || \hat{m}^{(a)}$
        else
            $M_a := M_a || 0^n$
for $i := 1 \dots q$ do
    $\mathsf{Tag}_i := \mathcal{O}(M_i)$
for $i := 1 \dots (q-1)$ do
    for $j := (i+1) \dots q$ do
        if $\mathsf{Tag}_i = \mathsf{Tag}_j$ then
            return 1
return 0
```

Description

We will use the following notation: given messages (i.e., attack queries) $M_1, \dots, M_q$ of length ℓ each, we denote the i-th block of the a-th message by $m_i^{(a)}$. We also analogously define $x_i^{(a)} := m_i^{(a)} \oplus p_i \cdot R$.

The adversary $\mathsf{A} := \mathsf{A}^{\mathcal{O}(\cdot)}_{\ell,q,n}$ we present is parametrized by variables ℓ, q, n (maximal length of messages, number of messages, size of message blocks), and expects to interact with an oracle $\mathcal{O}(\cdot)$ that is either PMAC, or a random function. Its pseudocode is given as Algorithm 2.

The adversary A first identifies the largest possible subset $S \subseteq P_{[\ell]} = (p_1, \dots, p_\ell)$ that is an additive subgroup of $GF(2^n)$; or more generally, a coset of any group H in G, where both H and G are additive subgroups of $GF(2^n)$ and do not need to be subsets of $P_{[\ell]}$. We denote the order of S by ℓ_S and the indices of S within $P_{[\ell]}$ by I_S. Additionally, we choose an arbitrary fixed element e in S. If S is a group, then for notational convenience we choose $e := 0$, but this is of no significance to the attack, or its proof. Then, we denote by I'_S the indices of $S \setminus \{e\}$ within $P_{[\ell]}$.

Having identified S, the adversary samples q message blocks $\hat{m}^{(1)}, \ldots, \hat{m}^{(q)} \overset{\$}{\leftarrow} \{0,1\}^n$ one by one, using a form of rejection sampling. Namely, it maintains a set

$$\mathcal{U}_{a-1} = \left\{ \frac{\hat{m}^{(b)} \oplus \hat{m}^{(c)}}{e \oplus p_i} \ : \ b, c \in [a-1], b \neq c, i \in I'_S \right\} ,$$

where a is the index of $\hat{m}^{(a)}$ currently sampled (intuitively, all $u \in \mathcal{U}_{a-1}$ have the property that if $R = u$, then $\mathsf{crCan}(M_b, M_c)$ for some $b \neq c \leq [a-1]$ occurs). A random value sampled for $\hat{m}^{(a)}$ is then accepted, only if the intersection

$$\left\{ \frac{\hat{m}^{(a)} \oplus \hat{m}^{(b)}}{e \oplus p_i} \ : \ b \in [a-1], i \in I'_S \right\} \cap \mathcal{U}_{a-1}$$

is not too large (more precisely, if it is not larger than roughly twice its expected value).

From the blocks $\hat{m}^{(1)}, \ldots, \hat{m}^{(q)}$, A constructs a set of queries by repeating the same block ℓ times:

$$\begin{aligned} M_1 &= (\hat{m}^{(1)})^\ell = \hat{m}^{(1)} || \hat{m}^{(1)} || \ldots || \hat{m}^{(1)} \\ M_2 &= (\hat{m}^{(2)})^\ell = \hat{m}^{(2)} || \hat{m}^{(2)} || \ldots || \hat{m}^{(2)} \\ &\ldots \\ M_q &= (\hat{m}^{(q)})^\ell = \hat{m}^{(q)} || \hat{m}^{(q)} || \ldots || \hat{m}^{(q)} \end{aligned}$$

and then replaces all the blocks of these newly created messages that correspond to indices not in I_S by an all-zero block (in fact, any block with fixed value would do).

From this point on, the attack is simple: A submits the messages constructed above as the attack queries; if there is a collision among the outputs of the oracle it outputs 1, otherwise it outputs 0.

Analysis

We first look at the running time of A. The only nontrivial part of it that is worth consideration is the loop on lines 7–9, which might potentially never terminate. However, note that the expected size of the set

$$\mathop{\mathbb{E}}_{\hat{m}^{(a)} \overset{\$}{\leftarrow} \{0,1\}^n} \left[\left| \left\{ \frac{\hat{m}^{(a)} \oplus \hat{m}^{(b)}}{e \oplus p_i} \ : \ b \in [a-1], i \in I'_S \right\} \cap \mathcal{U}_{a-1} \right| \right] \leq \frac{(a-1)^3(\ell_S - 1)^2}{2^n} ,$$

since each of the $(a-1)(\ell_S - 1)$ elements of the set intersected with $\mathcal{U}_{a-1}$ is individually uniform over $\{0,1\}^n$, and we are assessing the probability that it hits the set $\mathcal{U}_{a-1}$, where $|\mathcal{U}_{a-1}| \leq (a-1)^2(\ell_S - 1)$. Hence, the probability that a single iteration of the loop fails to satisfy the condition on line 9 is at most $1/2$ by Markov's inequality. Since every sampling on line 8 is independent, the probability (for a fixed a) that the loop is executed more than k times is upper bounded by 2^{-k}.

Now we move on to analyze the advantage achieved by our attack.

Theorem 7. *Let $P = (p_1, \dots, p_{2^n}) \in GF(2^n)$ be a progression as defined above, and let T_n be the mask distribution defined as $\tau_i = p_i \cdot R$ for a random $R \xleftarrow{\$} \{0,1\}^n$. Let Π, Π' be any distributions over $\mathcal{P}_n$ and assume that $\ell q^2 \le 2^{n-1}$. The adversary $\mathsf{A}_{\ell,q,n}$ given in Algorithm 2 achieves*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{\Pi,\Pi',\mathsf{T}_n}}(\mathsf{A}_{\ell,q,n}) \ge \frac{(\ell_S - 1)(q-1)^2}{2^{n+2}} - \frac{q^2}{2^n} \,,$$

where ℓ_S is the order of the largest coset S of some subgroup H in an additive subgroup G of $GF(2^n)$, such that the coset S is fully contained in $P_{[\ell]} = (p_1, \dots, p_\ell)$.

Note that as a special case, we can have $S = G$ and hence ℓ_S may be the order of the largest additive group contained in $P_{[\ell]}$.

Proof. We start by investigating the probability of $\mathsf{crCan}(M_a, M_b)$ for two distinct indices $a, b \in \{1, \dots, q\}$.

Lemma 17. *Let a, b be any two distinct indices from $\{1, \dots, q\}$. Then, we have*

$$\mathsf{Pr}\left[\mathsf{crCan}(M_a, M_b)\right] \ge \frac{\ell_S - 1}{2^n} \,.$$

Proof. (of Lemma 17) To slightly simplify the notation, we first prove the theorem for the case where S is a group and then describe the straightforward extensions needed to handle the case where S is a proper coset.

Let us hence assume that S is a group and therefore $e = 0$. We will denote by z the index of e in $P_{[\ell]}$, i.e., $p_z = e = 0$. For $i \in I'_S$, let r_i denote the value

$$r_i := \frac{\hat{m}^{(a)} \oplus \hat{m}^{(b)}}{p_i} \,, \tag{7.14}$$

where the division occurs in $GF(2^n)$ (recall that $i \in I'_S$, and hence $p_i \neq 0$). We observe that if R is sampled to equal r_i, we obtain

$$\hat{m}^{(a)} \oplus \hat{m}^{(b)} = R \cdot p_i = R \cdot (p_z \oplus p_i) \,, \tag{7.15}$$

and hence

$$\hat{m}^{(a)} \oplus p_z \cdot R = \hat{m}^{(b)} \oplus p_i \cdot R \,, \tag{7.16}$$

which is equivalent to $x_z^{(a)} = x_i^{(b)}$.

Moreover, we claim that if $R = r_i$, we obtain a complete cross-cancellation for M_a and M_b. To observe this, first note that the equation (7.16) also trivially implies $x_i^{(a)} = x_z^{(b)}$. Additionally, recall that we work in a field of characteristic 2, and hence the set $\{0 = p_z, p_i\}$ is a subgroup of S. Consequently, it induces a partition of S into $\ell_S/2$ cosets of the form $\{p_j, p_j \oplus p_i\}$, for $j \in I_S$. For a fixed j, let $k \in I_S$ be an index, such that $p_k = p_i \oplus p_j$ (there is a unique such index, since S is a group). For each coset $\{p_j, p_k\}$, we then obtain equalities $x_j^{(a)} = x_k^{(b)}$ and $x_k^{(a)} = x_j^{(b)}$, since (7.15) also implies

$$\hat{m}^{(a)} \oplus \hat{m}^{(b)} = R \cdot p_i = R \cdot (p_j \oplus p_i \oplus p_j) = R \cdot (p_j \oplus p_k) \,.$$

This is true for any $j \in I_S$ (hence, for all the $\ell_S/2$ cosets), implying a cross-cancellation.

Finally, note that for any $i \neq j$, we have $r_i \neq r_j$. This follows from equation (7.14), and the fact that S is a group. Hence, whenever R is sampled to take any of the $\ell_S - 1$ distinct values $\{r_i : i \in I'_S\}$, the event $\mathsf{crCan}(M_a, M_b)$ occurs, which concludes the proof for the case where S is a group.

Now assume that the set S is a proper coset of some subgroup H in a group $G \subseteq GF(2^n)$. Observe that $S = e \oplus H$, hence we can rewrite any element $p \in S$ as $p = e \oplus h$ for some $h \in H$ and vice versa, $h = e \oplus p$. For the sake of argument, imagine that the values $p_i \in S$ (note $S \subseteq P_{[\ell]} = (p_1, \ldots, p_\ell)$) on all positions in I_S would be replaced by $h_i := p_i \oplus g \in H$ instead; i.e., we would replace S by H in $P_{[\ell]}$ (recall that $|S| = |H|$). Then the previous analysis (for S being a subgroup) would apply, since H is a group. Now, if a cross-cancellation occurs in this modified setting with S replaced by H in $P_{[\ell]}$, then it also occurs before the replacement, as we have

$$\begin{aligned}\hat{m}^{(a)} \oplus p_i \cdot R = \hat{m}^{(b)} \oplus p_j \cdot R &\Leftrightarrow \hat{m}^{(a)} \oplus (p_i \oplus e) \cdot R = \hat{m}^{(b)} \oplus (p_j \oplus e) \cdot R\\ &\Leftrightarrow \hat{m}^{(a)} \oplus h_i \cdot R = \hat{m}^{(b)} \oplus h_j \cdot R\,.\end{aligned}$$

Hence, all the cancellations occur as before, even if we replace H by S in $P_{[\ell]}$, and the rest of the analysis remains the same. □

The above lemma shows that for each M_a, M_b there are at least $\ell_S - 1$ "good" values R can take that would cause a cross-cancellation for M_a and M_b. Interestingly, this holds even if M_a and M_b are constructed from arbitrary distinct fixed values $\hat{m}^{(a)}$ and $\hat{m}^{(b)}$.

Let us refer to these potential values of R as (a, b)-*good*, and let $\mathcal{R}_{a,b}$ denote the set of all (a, b)-good values, formally

$$\mathcal{R}_{a,b} = \{r \in \{0,1\}^n \,:\, (R = r) \Rightarrow \mathsf{crCan}(M_a, M_b)\}\ .$$

Let $\mathcal{R} = \bigcup_{a \neq b \in [q]} R_{a,b}$ denote the set of all good values.

We now need to show that when we look at all $\binom{q}{2}$ pairs of A's queries, most of these good values for R will not overlap, giving us $|\mathcal{R}| = \Omega(\ell_S q^2)$ in total. To this end, we leverage the rejection sampling that A used to choose the building blocks $\hat{m}^{(a)}$.

Lemma 18. *Assuming $\ell_S q^2 \leq 2^{n-1}$, we have*

$$|\mathcal{R}| \geq \frac{(\ell_S - 1)(q-1)^2}{4}\ .$$

Proof. (of Lemma 18) For $a \in [q]$, let $\mathcal{V}_a$ denote the set of fresh values that are added to the set $\mathcal{U}_{a-1}$ in the a-th iteration of step 10 of the algorithm $\mathsf{A}^{\mathcal{O}(\cdot)}_{\ell,q,n}$ to form the set $\mathcal{U}_a$, formally $\mathcal{V}_a := \mathcal{U}_a \setminus \mathcal{U}_{a-1}$. By the definition of $\mathcal{U}_a$ on line 10, and the fact that we only count fresh values, we have

$$\mathcal{V}_a = \left\{\frac{\hat{m}^{(a)} \oplus \hat{m}^{(b)}}{e \oplus p_i} \,:\, b \in [a-1], i \in I'_S\right\} \setminus \mathcal{U}_{a-1}\ .$$

The size of the set above before subtracting $\mathcal{U}_{a-1}$ is $(a-1)(\ell_S-1)$, and by the choice of $\hat{m}^{(a)}$ on lines 7–9, we know that the subtraction removes at most $2(a-1)^3(\ell_S-1)^2/2^n$ elements. Hence, we have

$$\begin{aligned} |\mathcal{V}_a| &\geq (a-1)(\ell_S-1) - \frac{2(a-1)^3(\ell_S-1)^2}{2^n} \\ &\geq (a-1)(\ell_S-1)\left(1-\frac{2(a-1)^2(\ell_S-1)}{2^n}\right) \\ &\geq \frac{(a-1)(\ell_S-1)}{2}\ , \end{aligned}$$

where the last inequality follows, since $a^2\ell_S \leq q^2\ell_S \leq 2^{n-1}$. Clearly, $\mathcal{U}_q = \bigcup_{a=1}^{q}\mathcal{V}_a$, and by construction the sets $\mathcal{V}_a$ are disjoint. Hence, we obtain

$$|\mathcal{U}_q| = \sum_{a=1}^{q}|\mathcal{V}_a| \geq \sum_{a=1}^{q}\frac{(a-1)(\ell_S-1)}{2} \geq \frac{(\ell_S-1)(q-1)^2}{4}\ .$$

Finally, by observations in the proof of Lemma 17, we have $\mathcal{U}_q \subseteq \mathcal{R}$. Therefore, we can also conclude that $|\mathcal{R}| \geq \frac{(\ell_S-1)(q-1)^2}{4}$. □

To conclude the proof of Theorem 7, note that when $\mathcal{O} = \mathsf{PMAC}$, and if the randomly sampled R takes any value from $\mathcal{R}$, A observes a tag collision and outputs 1. According to Lemma 18, this happens with probability at least $(\ell_S-1)(q-1)^2/2^{n+2}$. On the other hand, if $\mathcal{O}$ is a random function, A observes such a collision (and hence outputs 1) with probability at most $q^2/2^n$. □

Consider the Gray code used in the original PMAC construction. This code does not include the zero element, hence the progression $P = (p_1, \ldots, p_{2^n-1})$ in this case does not contain any additive groups. However, it does contain some proper cosets. To see this, let G_i denote the additive subgroup of $GF(2^n)$ of size 2^i containing elements of the form $0^{n-i}w$ for $w \in \{0,1\}^i$. Then for any $\ell \geq 2^k - 1$ we get that $P_{[\ell]}$ contains the only proper coset of G_{k-1} in G_k, which is of size 2^{k-1}. This gives us the following corollary.

Corollary 1. *Consider the setting from Theorem 7, and let $\mathtt{T}_n$ be the mask distribution defined as $\tau_i = \gamma_i^n \cdot R$ for a random $R \xleftarrow{\$} \{0,1\}^n$, and γ_i^n being the i-th codeword in the canonical Gray code. Then, we have*

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{PMAC}_{\Pi,\Pi',\mathtt{T}_n}}(\mathsf{A}_{\ell,q,n}) = \Omega(\ell q^2/2^n)\ .$$

Bibliography

[1] The keccak sponge function family. Accessed January, 10th, 2017; http://keccak.noekeon.org/.

[2] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[3] Joël Alwen, Peter Gaži, Chethan Kamath, Karen Klein, Georg Osang, Krzysztof Pietrzak, Leonid Reyzin, Michal Rolinek, and Michal Rybár. On the memory-hardness of data-independent password-hashing functions. Cryptology ePrint Archive, Report 2016/783, 2016. http://eprint.iacr.org/2016/783.

[4] Jee Hea An and Mihir Bellare. Constructing VIL-MACs from FIL-MACs: Message authentication under weakened assumptions. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 252–269. Springer, Heidelberg, August 1999.

[5] Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 602–619. Springer, Heidelberg, August 2006.

[6] Mihir Bellare, Daniel J. Bernstein, and Stefano Tessaro. Hash-function based PRFs: AMAC and its multi-user security. LNCS, pages 566–595. Springer, Heidelberg, 2016.

[7] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 1–15. Springer, Heidelberg, August 1996.

[8] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th FOCS*, pages 514–523. IEEE Computer Society Press, October 1996.

[9] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.

[10] Mihir Bellare and Anna Lysyanskaya. Symmetric and dual PRFs from standard assumptions: A generic validation of an HMAC assumption. Cryptology ePrint Archive, Report 2015/1198, 2015. http://eprint.iacr.org/2015/1198.

[11] Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved security analyses for CBC MACs. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 527–545. Springer, Heidelberg, August 2005.

[12] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006.

[13] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. UMAC: Fast and secure message authentication. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 216–233. Springer, Heidelberg, August 1999.

[14] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 384–397. Springer, Heidelberg, April / May 2002.

[15] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer Berlin Heidelberg, 2007.

[16] Chongwon Cho, Chen-Kuei Lee, and Rafail Ostrovsky. Equivalence of uniform key agreement and composition insecurity. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 447–464. Springer, Heidelberg, August 2010.

[17] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 430–448. Springer, Heidelberg, August 2005.

[18] Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and PRGs. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 649–665. Springer, Heidelberg, August 2010.

[19] Oxford Living Dictionaries. Cryptography. Accessed January, 17th, 2017; https://en.oxforddictionaries.com/definition/cryptography.

[20] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[21] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. Message authentication, revisited. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 355–374. Springer, Heidelberg, April 2012.

[22] Yevgeniy Dodis, Thomas Ristenpart, John P. Steinberger, and Stefano Tessaro. To hash or not to hash again? (In)differentiability results for h^2 and HMAC. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 348–366. Springer, Heidelberg, August 2012.

[23] Niels Ferguson. Collision attacks on ocb. *Preprint, February*, 2002.

[24] Peter Gaži, Krzysztof Pietrzak, and Michal Rybár. The exact PRF-security of NMAC and HMAC. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 113–130. Springer, Heidelberg, August 2014.

[25] Peter Gazi, Krzysztof Pietrzak, and Stefano Tessaro. Generic security of NMAC and HMAC with input whitening. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 85–109. Springer, Heidelberg, November / December 2015.

[26] Peter Gaži, Krzysztof Pietrzak, and Michal Rybár. The exact security of pmac. *IACR Transactions on Symmetric Cryptology*, 2016(2):145–161, 2017.

[27] Oded Goldreich. *Foundations of cryptography.* Cambridge University Press, Cambridge, UK New York, 2006.

[28] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

[29] G. H. Hardy and Edward M. Wright. *An Introduction to the Theory of Numbers (sixth edition).* Oxford University Press, USA, 2008.

[30] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, 1980.

[31] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 8–26. Springer, Heidelberg, August 1990.

[32] Dimitar Jetchev, Onur Özen, and Martijn Stam. Understanding adaptivity: Random systems revisited. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012*, volume 7658, pages 313–330, 2012.

[33] C. R. Jordan. *Groups.* Newnes, Oxford, 1994.

[34] D. Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet.* Scribner, 1996.

[35] J. Katz and Y. Lindell. *Introduction to modern cryptography, 2nd ed.* CRC Press/Taylor & Francis, Boca Raton, 2015.

[36] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient authentication from hard learning problems. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 7–26. Springer, Heidelberg, May 2011.

[37] Jongsung Kim, Alex Biryukov, Bart Preneel, and Seokhie Hong. On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1. In Roberto Prisco and Moti Yung, editors, *Security and Cryptography for Networks*, volume 4116, pages 242–256, 2006.

[38] Neal Koblitz and Alfred Menezes. Another look at HMAC. Cryptology ePrint Archive, Report 2012/074, 2012.

[39] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: Keyed-hashing for message authentication. IETF Internet Request for Comments 2104, February 1997.

[40] Gaëtan Leurent, Thomas Peyrin, and Lei Wang. New Generic Attacks against Hash-Based MACs. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, volume 8270, pages 1–20. 2013.

[41] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988.

[42] Atul Luykx, Bart Preneel, Alan Szepieniec, and Kan Yasuda. On the influence of message length in PMAC's security bounds. LNCS, pages 596–621. Springer, Heidelberg, 2016.

[43] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC mode for lightweight block ciphers. In *FSE 2016*, LNCS, pages 43–59. Springer, Heidelberg, 2016.

[44] Vadim Lyubashevsky and Daniel Masny. Man-in-the-middle secure authentication schemes from LPN and weak PRFs. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 308–325. Springer, Heidelberg, August 2013.

[45] Ueli Maurer. Conditional equivalence of random systems and indistinguishability proofs. In *2013 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 3150–3154, July 2013.

[46] Ueli M. Maurer. Indistinguishability of random systems. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer, Heidelberg, April / May 2002.

[47] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Heidelberg, February 2004.

[48] Ueli M. Maurer and Stefano Tessaro. Computational indistinguishability amplification: Tight product theorems for system composition. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 355–373. Springer, Heidelberg, August 2009.

[49] Kazuhiko Minematsu and Toshiyasu Matsushima. New bounds for PMAC, TMAC, and XCBC. In Alex Biryukov, editor, *FSE 2007*, volume 4593 of *LNCS*, pages 434–451. Springer, Heidelberg, March 2007.

[50] Yusuke Naito, Yu Sasaki, Lei Wang, and Kan Yasuda. Generic State-Recovery and Forgery Attacks on ChopMD-MAC and on NMAC/HMAC. In Kazuo Sakiyama and Masayuki Terada, editors, *Advances in Information and Computer Security*, volume 8231, pages 83–98. 2013.

[51] Mridul Nandi. A unified method for improving PRF bounds for a class of blockcipher based MACs. In Seokhie Hong and Tetsu Iwata, editors, *FSE 2010*, volume 6147 of *LNCS*, pages 212–229. Springer, Heidelberg, February 2010.

[52] Mridul Nandi and Avradip Mandal. Improved security analysis of pmac. *Journal of Mathematical Cryptology*, 2(2):149–162, 2008.

[53] Thomas Peyrin, Yu Sasaki, and Lei Wang. Generic Related-Key Attacks for HMAC. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012*, volume 7658, pages 580–597. 2012.

[54] Thomas Peyrin and Lei Wang. Generic Universal Forgery Attack on Iterative Hash-Based MACs. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441, pages 147–164. 2014.

[55] Krzysztof Pietrzak. Composition does not imply adaptive security. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 55–65. Springer, Heidelberg, August 2005.

[56] Krzysztof Pietrzak. Composition implies adaptive security in minicrypt. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 328–338. Springer, Heidelberg, May / June 2006.

[57] FIPS Pub. 198, the keyed-hash message authentication code (hmac). *Federal Information Processing Standards Publication*, 198, 2002.

[58] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, Heidelberg, December 2004.

[59] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. Ocb: A block-cipher mode of operation for efficient authenticated encryption. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 196–205. ACM, 2001.

[60] Stefano Tessaro. Security amplification for the cascade of arbitrarily weak PRPs: Tight bounds via the interactive hardcore lemma. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 37–54. Springer, Heidelberg, March 2011.

[61] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 17–36. Springer, Heidelberg, August 2005.

[62] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 19–35. Springer, Heidelberg, May 2005.

[63] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981.

[64] Kan Yasuda. “sandwich” is indeed secure: How to authenticate a message with just one hashing. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *ACISP 07*, volume 4586 of *LNCS*, pages 355–369. Springer, Heidelberg, July 2007.

[65] Kan Yasuda. A new variant of PMAC: Beyond the birthday bound. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 596–609. Springer, Heidelberg, August 2011.

[66] Kan Yasuda. PMAC with parity: Minimizing the query-length influence. In Orr Dunkelman, editor, *CT-RSA 2012*, volume 7178 of *LNCS*, pages 203–214. Springer, Heidelberg, February / March 2012.

[67] Fei Yu, Michal Rybar, Caroline Uhler, and Stephen E Fienberg. Differentially-private logistic regression for detecting multiple-snp association in gwas databases. In *International Conference on Privacy in Statistical Databases*, pages 170–184. Springer International Publishing, 2014.

[68] Yusi Zhang. Using an error-correction code for fast, beyond-birthday-bound authentication. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 291–307. Springer, Heidelberg, April 2015.