Cryptographic Constructions Supporting Implicit Data Integrity

Michael Kounavis, David Durham, Sergej Deutsch,
Antonios Papadimitriou and Amitabh Das

Intel Labs, Intel Corporation, 2111, NE 25th Avenue, Hillsboro, OR 97124
Email: {michael.e.kounavis, david.durham, sergej.deutsch,
antonios.papadimitriou, amitabh.das}@intel.com

May 2018

Abstract

We study a new methodology supporting data integrity called 'implicit integrity' and present cryptographic constructions supporting it. Implicit integrity allows for corruption detection without producing, storing or verifying mathematical summaries of the content such as MACs, ICVs etc. The main idea behind this methodology is that, whereas typical user data demonstrate patterns such as repeated bytes or words, decrypted data resulting from corrupted ciphertexts no longer demonstrate such patterns. Thus, by checking the entropy of some decrypted ciphertexts, corruption can be possibly detected.

We discuss that implicit integrity can be associated with a notion of security which is different from the typical requirement that the output of cryptographic systems should be indistinguishable from the output of a random permutation. The notion of security we discuss is that it should be computationally difficult for an adversary to corrupt some ciphertext so that the resulting plaintext demonstrates specific patterns. We further introduce two kinds of adversaries. First, an input perturbing adversary performs content corruption attacks. Second an oracle replacing adversary performs content replay attacks. We discuss requirements for supporting implicit integrity in these two adversary models, and provide security bounds for a construction called IVP, a three-level confusion diffusion network which can support implicit integrity and is inexpensive to implement.

1 Introduction

1.1 The Concept of implicit data integrity

This paper addresses the problem of corruption detection without producing, storing or verifying mathematical summaries of the content. Such summaries, typically known as Message Authentication Codes (MACs) [3] [4] or Integrity Check Values (ICVs) are typically costly to maintain and use. The standard way of supporting data integrity is by using MACs produced by cryptographic hash functions such as SHA256 [1] or SHA3 [2], the use of which typically results in latency, storage and communication overheads. These overheads are due to the unavoidable message expansion associated with using the MACs. For example, if a system protects a cache line with a MAC [7], this MAC value needs to be read in every data read operation. This wastes memory access bandwidth resources as each data read operation needs to be realized as two memory read operations.

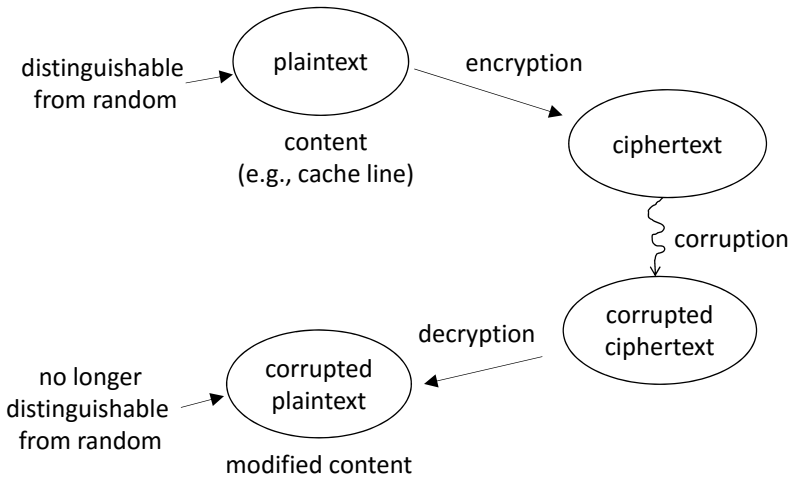


Figure 1: The concept of implicit integrity

Another area where our work applies is in network and communication systems which employ a number of different protocols at different layers of the network stack, ranging from the link layer to the application layer, with additional corruption detection mechanisms [3] [4] [24]. Many of these protocols employ corruption detection mechanisms to cater for any intentional or unintentional corruptions encountered during the transmission of data from point A to point B. The cost of providing such capabilities is the additional metadata per packet, which can be sigificant. This paper describes an alternative methodology that uses pattern techniques in order to support corruption detection for the large majority of user data without message expansion. The main idea is discussed below.

If some content exhibits patterns (i.e., has low entropy), then such content can be distinguished from random data. Let's consider that this content is encrypted, as shown in the figure, where the encryption algorithm is a good pseudo-random permutation and thus can successfully approximate a random oracle. The cipher text which is produced in this way is no longer distinguishable from random data, under certain reasonable assumptions about the adversary. Any corruption on the cipher text results in a new cipher text value, which is different from the original one. Furthermore, any decryption operation on this new cipher text value results in a corrupted plaintext value which is different from the original one as well. As decryption is the inverse operation of encryption, the decryption algorithm also approximates a random oracle under reasonable adversary models. Because of this reason, the corrupted plaintext value is also indistinguishable from random data with very high probability. From a system realization standpoint, the corrupted plaintext is indistinguishable from random data because block ciphers or cryptographic constructions typically perform strong mixing of their input bits. Due to an 'avalanche effect' associated with the decryption oracle, even a single bit change in the cipher text can affect all bits of the decrypted plaintext. For these reasons, checking the entropy of the result of a decryption operation can be a reliable test for detecting corruption for some data. We refer to such methodology as 'implicit data integrity' or just 'implicit integrity'.

One of the main challenges in building a system, the operation of which is based on the principle of implicit integrity is how to define 'high or 'low' entropy. It is not straightforward how to determine that some content's entropy is 'low enough' or 'high enough' so as to safely deduce that the original content has not been corrupted. Another challenge has to do with

the design of cryptographic constructions for supporting implicit integrity and the derivation of analytical proofs associated with security claims. A large body of work exists that focuses on bounds associated with distinguishing the output of cryptographic constructions from that of a random permutation. In our work we argue that such strong notion of security may not be required in the context of implicit integrity. The security objective which we discuss in this paper is that it should be computationally hard for adversaries to modify a set of given ciphertext messages so as to result in plaintexts of low entropy, while preserving confidentiality. This paper focuses on formalizing this security objective, exploring the requirements of cryptographic constructions supporting it, and providing an example of a simple, implementable three level confusion diffusion network that meets these requirements for a specific security level.

1.2 Summary of our contributions

A first contribution of this paper is the introduction of a class of constructions which we refer to as 'Random Oracles according to Observer functions' (RO^2). An observer function is a function that searches the output of cryptographic systems in order to detect unusual behavior, such as the presence of patterns. Patterns can be repeated nibbles, bytes, words or double words. If an observer function detects unusual behavior with the same or similar probability in a cryptographic system's output as in a random oracle's output then such cryptographic system belongs to the class of RO^2 constructions associated with the specific observer function.

A second contribution of this paper is the description of a security model against data corruption and replay attacks, which is associated with implicit integrity and connected with the class of RO^2 constructions. Specifically, we show that if a construction is in the RO^2 class, then the construction always supports some form of implicit data integrity, and is secure in the proposed model. The proposed model comprises two kinds of adversaries. First, an input perturbing adversary is an algorithm which is given a set of ciphertext messages $q_0, \ldots, q_{m-1}$, the plaintexts of which exhibit patterns and a query bound B. The algorithm succeeds if it finds a ciphertext message y which is different from $q_0, \ldots, q_{m-1}$, the plaintext of which also exhibits patterns. Security in this adversary model indicates protection against data corruption attacks. Second, an oracle replacing adversary is an algorithm which is also given a set of cipher messages $q_0, \ldots, q_{m-1}$ the plaintexts of which exhibit patterns and a query bound B. The algorithm succeeds if it replaces a set of oracles $R_0, R_1, \ldots$ queried by the decryption system with a new set of oracles $R'_0, R'_1, \ldots$, so that there exists a ciphertext message $y \in \{q_0, \ldots, q_{m-1}\}$ the plaintext of which continues to exhibit patterns even when the oracles $R_0, R_1, \ldots$ of the decryption system are replaced by $R'_0, R'_1, \ldots$. We discuss that security in this adversary model indicates protection against content replay attacks.

A third contribution of this paper is a cryptographic construction called IVP (stands for 'Implicit integrity Via Pre-processing'), which is in the proposed class RO^2. IVP is a three level confusion diffusion network that includes pseudo-random permutations of varying widths. Proofs for this construction derive from computing the probability that the system state is being altered by the flow of differentials inside the construction. Differentials are caused by adversarial perturbations. In our analysis we consider that the number of queries issued by adversaries is bounded in such a way, so that the internal pseudo-random permutations queried by the proposed construction are indistinguishable from truncated output random oracles. In this case, differential signals coming out of these components are indistinguishable from random, uniformly distributed and statistically independent signals. The methodology followed in the proofs is similar to the methodology used for searching for zero correlation linear hulls in block ciphers [19].

1.3 Protecting data that do not exhibit patterns

One issue that needs addressing, when building a system that protects data using the principle of implicit integrity, is how to detect corruption if data do not exhibit patterns. As we elaborate below, patterns can be found in up to 91% of the data of client workloads and 84% of the data of server workloads. The numbers come from observations on 111 million representative client workload cache lines and 1.4 billion representative server workload cache lines. Whereas such data can be protected using implicit integrity, the remaining $9\% - 16\%$ of the data need protecting also.

Such design issue can be easily addressed. Implementations can protect the overwhelming majority of user data that exhibit patterns using implicit integrity and the remaining data using standard techniques. There is nothing in the implicit integrity methodology, discussed here, that prevents it from being used together with other independent integrity mechanisms such as MACs. Such solutions can co-exist with implicit integrity.

Cost reduction comes from the fact that only MACs computed from high entropy data need to be stored and accessed. If some decrypted content exhibits patterns, then there is some assurance that no corruption has occurred. If no patterns are exhibited, however, a search can be made for a MAC associated with the content. If no MAC is found, then the data is deemed corrupted. Otherwise, an integrity check is made using the returned MAC. Such implementation can use a content addressable memory unit or a hash table for accessing and managing MACs. We remind that MACs are significantly fewer than those required for protecting all the data, and this is the main advantage of this approach. Further investigations on hardware and operating system changes required in order to support implicit integrity are beyond the scope of this paper.

2 Related work

We are the first to analyze the concept of implicit integrity and are not aware of any other published work that proposes or studies any similar integrity methodology. Besides this, the concept of random oracles according to observer functions bears some similarity with the concept of correlation intractability discussed in reference [20]. The aim of reference [20] is different from ours. Reference [20] establishes that there exist signature and encryption schemes that are secure in the random oracle model, but for which any implementation of a random oracle results in insecure schemes. In the process, reference [20] introduces two concepts which are related to our definition of the class RO^2. First, a binary relation is introduced as 'evasive' if when given access to a random oracle O, it is infeasible to find a string x so that the pair $(x; O(x))$ is in the relation. Second, a function ensemble F is called correlation intractable if for every evasive binary relation, given the description of a uniformly selected function f_s from F it is infeasible to find an x such that $(x; f_s(x))$ is in the relation.

Notable differences between our paper and [20] are the following: First, the binary relation as defined in [20] involves a relationship between two strings. In our case, we employ an observer function which examines the output of the system and produces a Boolean value from this output, after running a polynomial time algorithm. The two definitions may be equivalent, however, the difference in the formalism reflects further contrasts. Second, in reference [20], much is said about intractability. There is a function defined, which characterizes the probability values associated with the output of a system as negligible or non-negligible, and the whole function ensemble as tractable or intractable. This is a 'black-and-white' binary characterization which is not directly relevant to our analysis. In the case of the RO^2 definition, we are more interested in the relative behavioral difference in the output

of a system, when compared to a random oracle or a random function. For this purpose, we introduce an indistinguishability parameter ϵ. This parameter characterizes this relative behavioral difference between the real system we examine and a random oracle or random function primitive.

Third, we examine only systems associated with query bounds. Essentially, we push forward research on open problems identified in [20] and associate solutions to these problems (i.e., correlation intractability in the presence of adversary query bounds) with practical implications such as the ability to detect corruptions without message authentication tags. As part of the process, we discover that it is not possible to support implicit integrity without introducing a number of structural constraints in the systems examined. These structural constraints are a unique contribution of our work, and are sufficient in order to demonstrate the usefulness of implicit integrity systems against data replaying adversaries.

In our work we also make extensive use of the results of references [11] [12] which allow us to consider the internal pseudo-random permutations of our constructions as truncated output random oracles. Finally, a large body of work exists on wide block ciphers and wide block cipher constructions, which could be used instead of the constructions proposed here for supporting implicit integrity. The main difference between our proposed constructions and wide block ciphers is in the security objective which guides the system designs. Wide block ciphers and similar constructions are designed with the aim of being indistinguishable from random permutations or truncated output random oracles. Implicit integrity systems are designed with a different aim in mind: That of being computationally difficult to manipulate so as to create ciphertexts the plaintexts of which exhibit patterns. Such difference in objectives is reflected in the complexity of the system designs. For example, in section 6 we propose a simple three-level confusion diffusion network which provides some assurances against data corruption and data replay attacks without compromising the confidentiality of the output, and which is structurally simpler than some wide block cipher constructions (e.g., [26]). We come back to this issue later in the paper.

3 Preliminary concepts and definitions

3.1 Random Oracles according to Observer functions (RO^2)

We begin our discussion with the concept random oracles according to observer functions. We consider 'observer functions' that search cryptographic system outputs in order to detect 'abnormal behavior', such as repetitions of values of different sizes. Such repetitions are not different from the repetitions termed "needles in a haystack", which are studied in reference [23].

The concept is illustrated in Figure 2. Function f observes unusual behavior in the values of the output of a random oracle with probability P_f. The same function f observes the same unusual behavior in the values of the output of the real system S with probability $P_{f,2}$. If $P_{f,2} \leqslant P_f \cdot 2^{\epsilon}$ for a given maximum *non-repeating* input sequence of size B, and if this relation holds even when $P_{f,2}$ is conditioned upon previous inputs, then the system S is a "random oracle according to observer function f" associated with an indistinguishability parameter ϵ: $S \in \mathsf{RO}^2(f, B, \epsilon)$.

The query bound B denotes the life time of the construction. Furthermore, the query bound B is related to additional structural constraints discussed below. A finite life time B is introduced in the definition of a RO^2 construction, so that the construction can contain primitives which are bijective functions (pseudo-random permutations) and which within the bounds of the life time B are indistinguishable from truncated output random oracles. In what

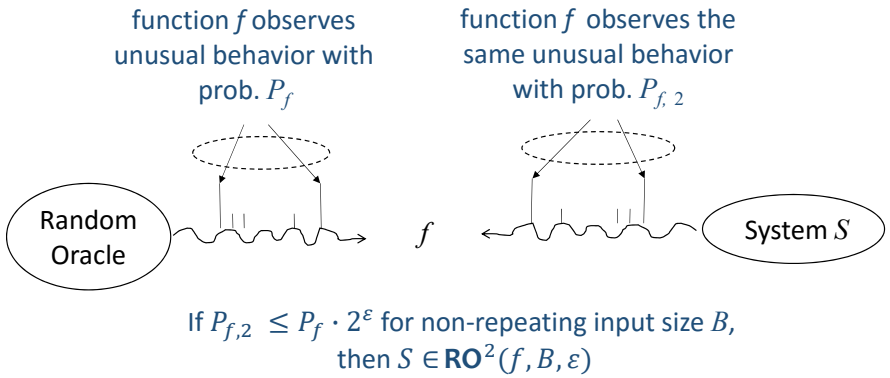


Figure 2: The Concept of a Random Oracle according to an Observer function (RO^2)

follows we will be denoting such primitives as 'ingredient random permutations'.

In order for a construction to be RO^2 it needs to satisfy the condition $P_{f,2} \leqslant P_f \cdot 2^{\epsilon}$ for non-repeating input. So what does non-repeating input mean? Having non-repeating input means that:

- within a lifetime of a construction $\{y_0, y_1, \ldots, y_{B-1}\}$ input is not repeating, i.e., $y_i \neq y_j \; \forall i, j \in [0, B-1]$; and
- inputs that result in unusual behavior in one lifetime $\{y_0, y_1, \ldots, y_{B-1}\}$ of a construction are not repeated in any other lifetime $\{z_0, z_1, \ldots, z_{B-1}\}$ of the same construction.

Whereas the inputs to constructions that are in the class RO^2 can be from any set of values, the constructions are most useful when the inputs considered are adversary queries, i.e., corrupted ciphertext values. In this case, the conditions of non-repeating input are not as restrictive as they sound. The intuition behind introducing these conditions, which are used in the derivation of our main results below, is that if an adversary repeats queries as part of the attack strategy, then the system provides the same output for these repeated queries. Specifically, we consider that there are no parameters, potentially randomizing the system which are out of the adversary's control. Under this assumption it is clear that it is not beneficial for an adversary to repeat queries which are unsuccessful, as their result will be the same. On the other hand, if some queries are successful, then the adversary does not need to repeat these queries, as the adversary possesses the knowledge about the impact of such queries. Because of these reasons, it is not restrictive to introduce the non-repeating input requirement, as we consider adversaries that do not repeat their queries to the construction. We also note that we do not restrict every input to the construction. We just restrict only the inputs for which the condition $P_{f,2} \leqslant P_f \cdot 2^{\epsilon}$ needs to be satisfied.

Definition 1: *Observer function.* A function f associated with input strings of length L, $f : \{0,1\}^L \to \{0,1\}$ is called an observer function if it outputs only one of two values 0 or 1. If for some input x, $f(x) = 1$, then we will be saying that input x demonstrates unusual behavior according to observer function f. We will also be denoting this fact as $x \in \mathbf{\Pi}(f)$.

Definition 2: *Random Oracle according to an Observer function* (RO^2). Let $\{y_0^{(0)}, y_1^{(0)}, \ldots, y_{m_0-1}^{(0)}\}$, $\{y_0^{(1)}, y_1^{(1)}, \ldots, y_{m_1-1}^{(1)}\}$, $\ldots$ be sets of binary strings of length L, the cardinalities of which satisfy $m_i \leqslant B$, $\forall i \geqslant 0$. Let also f be an observer function associated with inputs

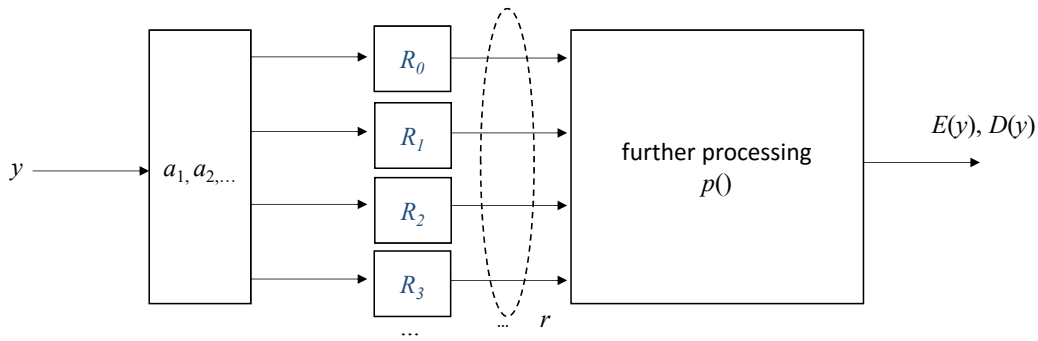


Figure 3: Internal structure of a constrained RO^2 construction

of length L, and $R \leftarrow 2^\infty$ a random oracle. A function or system $S : \{0,1\}^L \rightarrow \{0,1\}^L$ is called a random oracle according to observer function f associated with a life time B and indistinguishability parameter ϵ if the following conditions are true:

i. $y_i^{(k)} \neq y_j^{(k)}$ for all $y_i^{(k)}, y_j^{(k)} \in \{y_0^{(k)}, y_1^{(k)}, \ldots, y_{m_k-1}^{(k)}\}$ such that $i \neq j$, $k \geqslant 0$;

ii. if $y \in \mathbf{\Pi}(f)$ and $y \in \{y_0^{(k)}, y_1^{(k)}, \ldots, y_{m_k-1}^{(k)}\}$ for some $k \geqslant 0$, then $y \notin \{y_0^{(l)}, y_1^{(l)}, \ldots, y_{m_l-1}^{(l)}\}$ for all $l \neq k$;

iii. for all inputs y and collections of inputs $y_0, \ldots, y_{q-1} \geqslant 0$ such that $y \neq y_0, \ldots, y \neq y_{q-1}$, $y \in \{y_0^{(k)}, y_1^{(k)}, \ldots, y_{m_k-1}^{(k)}\}$, $y_i \in \{y_0^{(l_i)}, y_1^{(l_i)}, \ldots, y_{m_{l_i}-1}^{(l_i)}\}$, $k \geqslant 0$, $l_i \geqslant 0$, $0 \leqslant i < q$ and $q \geqslant 0$, the following is true:

$\text{Prob}[S(y) \in \mathbf{\Pi}(f) \mid y_0, \ldots, y_{q-1}] \leqslant P_f \cdot 2^\epsilon$ where $P_f = \text{Prob}[trunc_L(R(y)) \in \mathbf{\Pi}(f)]$

where the function $trunc_L()$ used in Definition 2 truncates its input returning the input's L most significant bits.

When a system S is a random oracle according to an observer function f, life time B and indistinguishability parameter ϵ, we will be denoting this fact as $S \in \mathbf{RO}^2(f, B, \epsilon)$. We also note that condition (iii) in Definition 2 covers all cases where the probability of seing unusual behavior in the output of S is conditioned upon any set of input values different from y of cardinality q. In the special case were $q = 0$ (i.e., no conditioning) this third condition is simplified as $\text{Prob}[S(y) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^\epsilon$. The probability value P_f will be denoted as 'observation probability' or 'pattern observation probability' associated with observer function f in this document.

A 'constrained' RO^2 function or system uses a number of internal invertible functions which are random permutations and which, for the life time (i.e., query) bound B used, are practically indistinguishable from random oracles in both processing directions. These are the primitives referred to as ingredient random permutations. An RO^2 system is invertible itself. One direction is denoted as E and referred to as 'encryption', whereas the other direction is denoted a D and referred to as 'decryption'. A constrained RO^2 system accepts a construction input y and uses a set of pre-processing polynomial time algorithms $a_1, a_2, \ldots, a_n$ to compute the inputs to ingredient random permutations which provide a response vector r. Then, it is this response which is further used in the RO^2 system for processing. Processing is done by an invertible polynomial time algorithm $p()$ and the input y is no longer used. A constrained RO^2

system is illustrated in Figure 3. In what follows, whenever we will be using the term RO^2 we will be referring only to constrained RO^2 systems.

In what follows we will be using the notation $E^{R_0,R_1,R_2,\ldots}$ and $D^{R_0,R_1,R_2,\ldots}$ to refer to encryption and decryption systems (encryption and decryption oracles) with access to the ingredient random permutations $R_0, R_1, R_2, \ldots$. Furthermore if $E^{R_0,R_1,R_2,\ldots}$ and $D^{R_0,R_1,R_2,\ldots}$ are RO^2 associated with an observer function f, a query bound B and an indistinguishability parameter ϵ, we will be denoting this fact as:

$$E^{R_0,R_1,R_2,\ldots}, D^{R_0,R_1,R_2,\ldots} \in \mathbf{RO}^2(f, B, \epsilon) \tag{3.1}$$

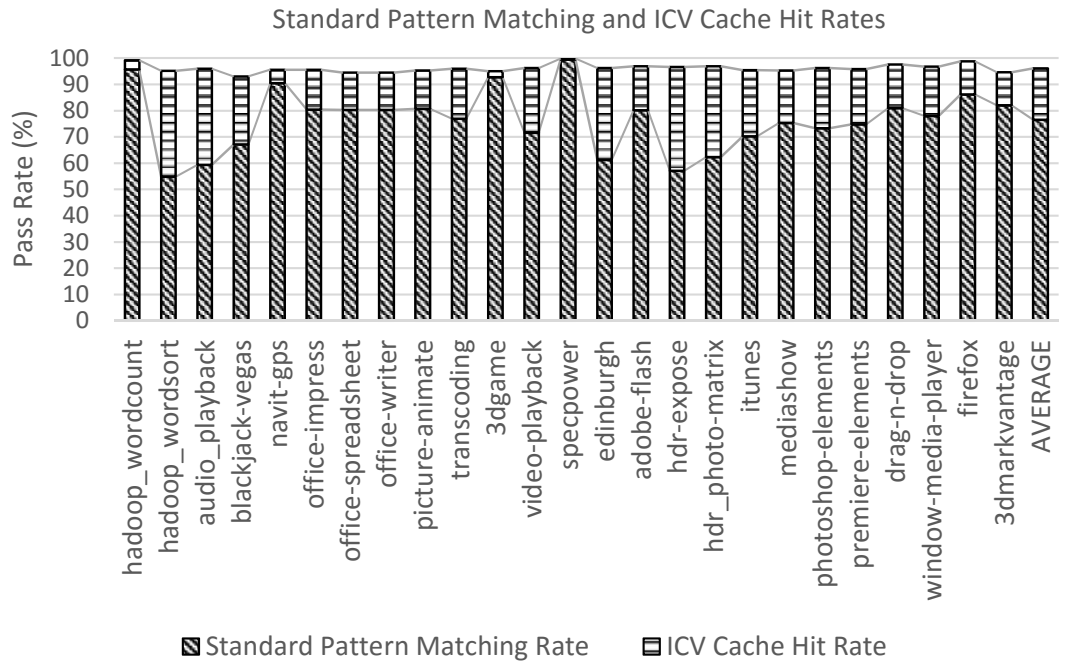
Finally, regarding the construction of Figure 3, we note that if algorithms $a_1, a_2, \ldots, a_n$ are replaced by the identity function and the data path of the figure implements a decryption operation, then the ciphertext of such system is obtained directly from the output of random permutations $R_0, R_1, R_2, \ldots$. This means that the system does not compromise the confidentiality offered by $R_0, R_1, R_2, \ldots$ in any way, provided that the implementation of $p()$, does not use any secret information also used by the implementations of $R_0, R_1, R_2, \ldots$. Setting $a_1, a_2, \ldots, a_n$ to the identity function is a choice we made in the design of IVP discussed in Section 6.

3.2 Pattern frequency observers

By 'Pattern Frequency Observers' (PFO) we mean observer functions that output true if a number of values that are equal to each other, from a given input set, exceed a threshold. Value types can range and may include nibbles, bytes, words and double words. The reason why we study these functions in this paper, is because such functions have experimentally been proven successful in characterizing the overwhelming majority of typical uncompressed, unencrypted client and server data. We have observed that client and server data demonstrate repetitions of values of different types, which in truly random data (i.e, random oracle outputs) appear with very low probability. It is these observations that motivate the implicity integrity work.

The fact that uncompressed, unencrypted user data demonstrate patterns should not come as a surprise. User data often consist of code, data structures, media data, pointer tables, and other types of structured data that are characterized by significant redundancy. For example, there exists a simple pattern which is frequently encountered in client and server data. This is the appearance of 4 or more 16-bit words which are equal to each other in a collection of 32 words. In this pattern the input size is 512 bits (i.e., each data is a memory cache line). An observer function which detects the presence of such pattern in inputs of size $L = 512$ bits is denoted as '$f_{4\times16}$'. Our experimental observations come from over 111 million client cache lines and 1.47 billion server cache lines of typical workloads. According to these observations, the $f_{4\times16}$ pattern characterizes 82% of the client cache lines and 78% of the server cache lines.

Pass rate comparisons associated with different pattern detectors are shown in Figure 4. Pattern detection based on the observer function $f_{4\times16}$ is referred to as 'Standard Pattern Matching' (SPM). Pattern detection based on the equality of words as well as other types of data such as bytes, double words and nibbles is referred to as 'Extended Pattern Matching' (EPM). As is evident from the figure, there are many typical client workloads (e.g., Microsoft® Office, transcoding, video player), the pass rates of which are quite high, ranging between 75%-80%, when Standard Pattern Matching is employed. These pass rates are boosted to 98% when Extended Pattern Matching is employed. Overall, the average client pass rates associated with Standard Pattern Matching are 82%. The average pass rates associated with Extended Pattern Matching are 91%. For server data, the corresponding pass rates are 78% and 84% respectively.



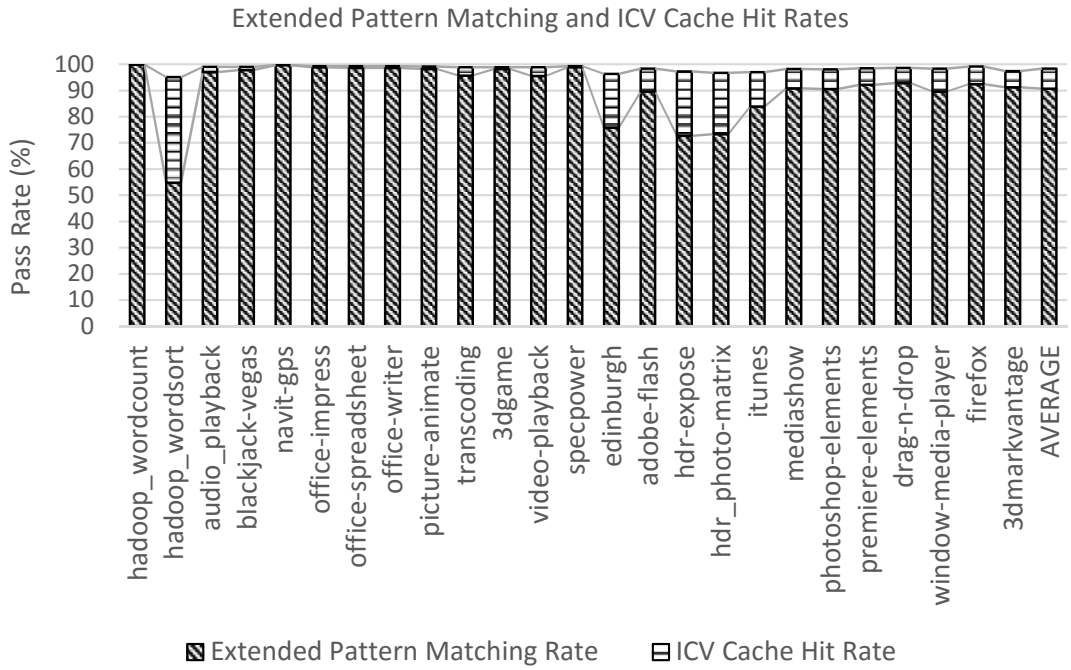


Figure 4: Pass rates associated with Standard and Extended Pattern Matching on client workload cache lines

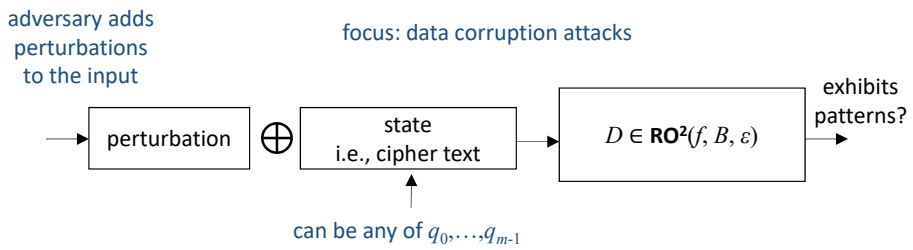


Figure 5: Input perturbing adversary

The EPM scheme encompasses many more pattern detectors than SPM. Pattern frequency observers, associated with the EPM scheme detect entities among the input data which are not only equal to each other, but are also placed in continuous index positions. These observers are not necessarily the same as those detecting value equalities. One can associate these two types of observer functions with different thresholds and, by doing so, build two different integrity schemes. Yet another type of observer function detects entities that take special values. Special values are values that are frequently encountered in regular user data but are infrequently encountered in random or corrupted plaintext data. For example, in memory cache lines obtained from client and server data workloads, a high percentage of bytes take the values of 0x00 or 0xFF.

A last type of observer functions used by EPM detects entities the value histogram of which demonstrates a sum of n highest entries (i.e., frequencies) being higher than a threshold. The intuition behind this type of pattern check is that there are several types of input messages, the content of which is not as random as that of encrypted data, but also does not demonstrate patterns at the byte or word granularity. One example of such content is media data, where nibble values may be replicated, but data do not demonstrate significant byte or word replications. Our experimental studies showed that there are millions of cache lines demonstrating a limited set of byte equalities but many nibble equalities. By checking whether the sum of the two highest nibble frequencies exceeds a threshold, a more flexible pattern detector can be built, which on the one hand encompasses significantly more regular user inputs, and on the other hand is associated with an event that is infrequent among random data.

Figure 4 also shows Integrity Check Value (ICV) cache hit rates for these algorithms. It is assumed that not all memory cache lines are protected visa implicit integrity, as discussed above, but some are protected using standard methods. For these, ICVs or MACs are cached. The hit rates of Figure 4 are obtained from a simulator implementing the caching of 32-bit ICVs using a 4KB cache unit for this purpose. When ICVs are cached the total pass rate observed by SPM and EPM are boosted significantly. For Standard Pattern Matching, the total pass rate, which includes a pattern check pass rate and an ICV cache hit rate, is 97%. For Extended pattern matching this rate is increased to 99%.

4 Adversary models and main security claims

4.1 Input perturbing adversary

Having presented some preliminary concepts and definitions, we are now in a position where we can present our adversary models and main security claims. The first type of adversary presented, describes adversaries which aim in corrupting encrypted data that are stored somewhere, or are in transit, in such a way so that the corruptions pass undetected. We refer to

such type of adversary as 'input perturbing' adversary.

The input perturbing adversary models any software process or physical intruder that aims in intentional corruptions of data. The underlying assumption of this adversary model is that the adversary can access data only in their encrypted form. This adversary can corrupt ciphertext data in any possible way hoping that the corruptions will result in plaintexts with patterns, and thus pass undetected. This adversary model is not unrealistic. In secure network connections or encrypted storage systems, many attacks originate from sources outside of these trusted domains (e.g., malware running in different processes or virtualized environments, untrusted hypervisors, man-in-the-middle attackers intercepting the packets of secure connections etc.) These attackers can possibly inspect a range of encrypted data, such as the whole encrypted memory of a computing system, but do not have access to the encryption keys required for obtaining the corresponding plaintexts. Thus, these attackers are unable to corrupt user data in a straightforward manner, such as changing special values from 0x00 to 0xFF. Attackers can only do this through the modification of ciphertexts, where ciphertexts are produced using keys unknown to the attackers.

More formally, the input perturbing adversary is defined as follows: Let's consider a pair of encryption and decryption oracles E and D such that $D = E^{-1}$ and for which $D, E \in \mathsf{RO}^2(f, B, \epsilon)$ for some f, B, ϵ. An input perturbing adversary $M^D(q_0, \ldots, q_{m-1}, B)$ is defined as a polynomial time algorithm which:

- has oracle access to D
- has knowledge of m queries $q_0, \ldots, q_{m-1}$ to D and their responses $D(q_0), \ldots, D(q_{m-1})$, where the responses exhibit patterns: $D(q_0), \ldots, D(q_{m-1}) \in \mathbf{\Pi}(f)$
- can perform at most B non-repeating queries to D as part of the game, which are other than $q_0, \ldots, q_{m-1}$.

The algorithm succeeds if it finds a new input data word y which is different from $q_0, \ldots, q_{m-1}$, the output of which exhibits patterns, i.e., $D(y) \in \mathbf{\Pi}(f)$. The number of query-response values known m is assumed not to be large enough so as to leak information about the internals of E, D allowing, for instance, rainbow table attacks. The input perturbing adversary type is shown in Figure 5. One special case of attacks performed by this adversary but not the only one studied in this paper, includes attacks where the query budget B is tight. These are the cases of on-line attacks where a single failure from the adversary's side exposes the attack (e.g., B can be 2^{32}). These attacks are further discussed in section 6.

The advantage of the input perturbing adversary is defined for the decryption operation as:

$$\begin{aligned}\mathbf{Adv}(M^D(q_0, \ldots, q_{m-1}, B), f) = \mathrm{Prob}[y \leftarrow M^D(q_0, \ldots, q_{m-1}, B);\\ y \notin \{q_0, \ldots, q_{m-1}\};\ D(y) \in \mathbf{\Pi}(f)]\end{aligned} \tag{4.1}$$

4.2 Oracle replacing adversary

Another type of adversary, the 'oracle replacing adversary' is associated with replay attacks. Replay attacks may happen across key domains such as network sessions that are encrypted with different keys, or encrypted memory domains. The model of the oracle replacing adversary can indeed be associated with such replay attacks under the assumption that the ingredient random permutations of an RO^2 construction use key values which are specific to particular

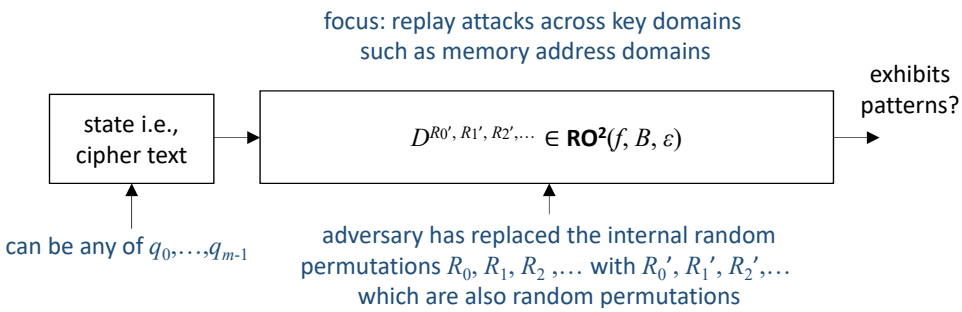


Figure 6: Oracle replacing adversary

domains of trust. When some valid encrypted data from one domain is replayed in another domain, then this replay attack is equivalent to replacing the ingredient random permutations of a RO^2 construction with another set of permutations, associated with a new domain, and observing the output.

More formally, lets consider a pair of encryption and decryption oracles E and D, $D = E^{-1}$ for which $D, E \in \mathbf{RO}^2(f, B, \epsilon)$ for some f, B, ϵ and have access to ingredient random permutations $R_0, R_1, R_2, \ldots$. An oracle replacing adversary $M^D(q_0, \ldots, q_{m-1}, B, \boldsymbol{R})$ is defined as a polynomial time algorithm for which the following are true:

- the algorithm has oracle access to $D^{R_0, R_1, R_2, \ldots}$
- the algorithm has knowledge of m queries $q_0, \ldots, q_{m-1}$ to $D^{R_0, R_1, R_2, \cdots}$ and their responses $D(q_0), \ldots, D(q_{m-1})$, where the responses exhibit patterns: $D(q_0), \ldots, D(q_{m-1}) \in \mathbf{\Pi}(f)$
- the algorithm has access to a set $\boldsymbol{R}$, $\boldsymbol{R} = \{\{R_0^{(0)}, R_1^{(0)}, \ldots\}, \ldots, \{R_0^{(n-1)}, R_1^{(n-1)}, \ldots\}\}$ of n sets of random permutations that have the same input and output length characteristics as $R_0, R_1, R_2, \ldots$ and are all different from $R_0, R_1, R_2, \ldots$
- can perform at most B non-repeating queries to $D^{R_0, R_1, R_2, \cdots}$ as part of the game, other than $q_0, \ldots, q_{m-1}$

The algorithm succeeds if it finds a set of new ingredient random permutations $\{R_0', R_1', R_2', \ldots\} \in \boldsymbol{R}$ and an input query word $y \in \{q_0, \ldots, q_{m-1}\}$, the output of which exhibits patterns when y is applied on an instance of D that uses the new ingredient random permutations $R_0', R_1', R_2', \ldots$, i.e., $D^{R_0', R_1', R_2', \cdots}(y) \in \mathbf{\Pi}(f)$. As in the case of the input perturbing adversary, this adversary also does not have any knowledge about possible keys used by E, D. The oracle replacing adversary is shown in Figure 6.

The advantage of the oracle replacing adversary is defined for the decryption operation as:

$$\begin{aligned}\mathbf{Adv}(M^D(q_0, \ldots, q_{m-1}, B, \boldsymbol{R}), f) = \mathrm{Prob}[\{\{R_0', R_1', \ldots\}, y\} \leftarrow M^D(q_0, \ldots, q_{m-1}, B, \boldsymbol{R});\\ y \in \{q_0, \ldots, q_{m-1}\};\ \{R_0', R_1', \ldots\} \in \boldsymbol{R};\ D^{R_0', R_1', \cdots}(y) \in \mathbf{\Pi}(f)]\end{aligned} \tag{4.2}$$

5 Main results

Our main results connect the concept of a random oracle according to an observer function with security claims associated with the input perturbing and oracle replacing adversary models.

Theorem 1: *About the security of an* RO^2 *construction in the input perturbing adversary model.* Given a pair of encryption and decryption oracles E and D, $D = E^{-1}$, an observer function f, a query bound B, and an observation indistinguishability parameter ϵ such that: $D, E \in \mathbf{RO}^2(f, B, \epsilon)$ then for any input perturbing adversary $M^D(q_0, \ldots, q_{m-1}, B)$:

$$\mathbf{Adv}(M^D(q_0, \ldots, q_{m-1}, B), f) \leqslant P_f \cdot 2^\epsilon \tag{5.1}$$

We note that P_f is the pattern observation probability associated with observer function f.

Proof of Theorem 1: We need to show that for every input perturbing adversary:

$$\text{Prob}[y \leftarrow M^D(q_0, \ldots, q_{m-1}, B);\ y \notin \{q_0, \ldots, q_{m-1}\};\ D(y) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^\epsilon \tag{5.2}$$

We assume that an adversary M exists for which the relation 5.2 does not hold. This adversary can succeed in repeatedly producing messages the outputs of which exhibit patterns with probability $> P_f \cdot 2^\epsilon$. We will show that if such adversary exists then it is not possible for E, D to belong to the set $\mathbf{RO}^2(f, B, \epsilon)$, which contradicts our assumption.

The adversary M is a polynomial time algorithm which performs at most B queries to oracle D. At the end of its computations the algorithm returns one of the following three outputs: (i) a value y which, if passed to oracle D, produces an output which exhibits patterns. In this case, the algorithm is successful; (ii) a value y which, if passed to oracle D, produces an output which does not exhibit patterns. In this case, the algorithm is unsuccessful; (iii) an indication that the algorithm has halted before returning any value y. In this case, the algorithm is unsuccessful as well.

From algorithm M, one can easily construct an algorithm M' which invokes M and behaves in the following way: If M does not halt before returning a y value, M' returns the same y value which M returns. If, on the other hand, M halts before returning a y value, then M' selects a value y_r at random and returns this value. Furthermore, algorithm M' maintains a table of previously returned y_r values so that every time algorithm M halts, the algorithm M' returns a different y_r value. From the definition of M', it is evident that algorithm M' also succeeds in repeatedly producing messages the outputs of which exhibit patterns with probability $> P_f \cdot 2^\epsilon$. Furthermore, algorithm M' never halts but always returns some y value.

The next step of the proof considers two cases for algorithm M' shown in Figures 7 and 8 respectively. In this first case, the algorithm M' as well as the underlying M distinguish between queries made to D that assist in the computation of a returned y value and the proposal of a y value. Such distinction is supported by the control flow of algorithms M, M'. Because of such distinction, the oracle D, which the adversary M' accesses on the side for his own calculations can be different the oracle D which is being attacked. In fact, we consider that such oracle is an exact replica of the attacked oracle.

In the second case, the oracle which the adversary M' accesses is indeed the same as the oracle D under attack. In this second case, there is no distinction between queries assisting the adversary computations and proposals of a y value. Nor such distinction is supported by the control flow of algorithms M', M.

In the first case, the adversary M' performs B attacks which are different from each other. For these attacks the adversary makes all queries but the one corresponding to the computed y to the oracle D, which is available on the side for his calculations. The computed y values are passed to the oracle which is being attacked. The expected value of the ratio of successful queries to the attacked oracle over all queries satisfies simultaneously $> P_f \cdot 2^\epsilon$ due to the assumption that the attacker exists and is successful and $\leqslant P_f \cdot 2^\epsilon$ due to the fact that the

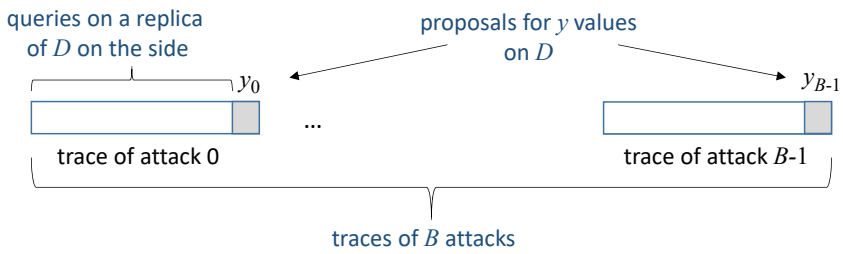


Figure 7: Hypothetical successful attacks of an input perturbing adversary M' (first case)

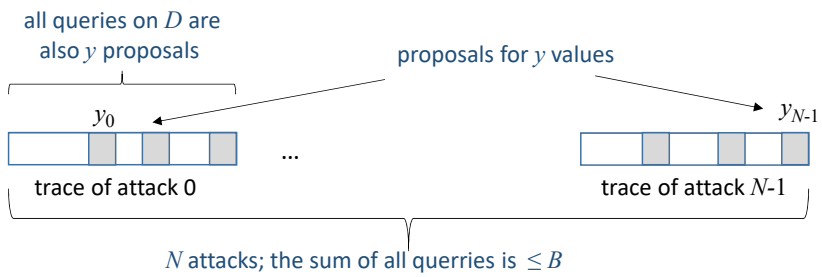


Figure 8: Hypothetical successful attacks of an input perturbing adversary M' (second case)

attacked oracle is $\mathbf{RO}^2(f, B, \epsilon)$ accepting non-repeating input of maximum length B, which is not possible.

In the second case, the adversary performs N attacks which are also different from each other. The sum of their trace lengths is at most B. In these attacks there is no distinction between queries assisting the adversary computations and proposals for a y value. The expected value of the ratio of successful queries to the attacked oracle over all queries again satisfies simultaneously $> P_f \cdot 2^\epsilon$ and $\leqslant P_f \cdot 2^\epsilon$ which is not possible. This is because we assume that the adversary exists, and that the attacked oracle is $\mathbf{RO}^2(f, B, \epsilon)$ accepting non-repeating input of maximum length B. Hence Theorem 1 is proven.

The reason for distinguishing between cases 1 and 2 in this proof is because such distinction allows us to construct different sequences of non-repeating inputs in each case, where these sequences demonstrate a paradox. Such sequences need to include quite many successful queries, due to the assumption that the adversary is successful, and at the same time quite too few due to the assumption that the construction used is $\mathbf{RO}^2(f, B, \epsilon)$.

A next theorem concerns the security of RO^2 constructions in the oracle replacing adversary model. It is in the proof of this theorem that we make use of the constraints of RO^2 constructions which we introduce in Figure 3 above.

Theorem 2: *About the security of an* RO^2 *construction in the oracle replacing adversary model.* Given a pair of encryption and decryption oracles E and D, $D = E^{-1}$, an observer function f, a query bound B, and an observation indistinguishability parameter ϵ such that: $D, E \in \mathbf{RO}^2(f, B, \epsilon)$ then for any oracle replacing adversary$M^D(q_0, \ldots, q_{m-1}, B, \boldsymbol{R})$:

$$\mathbf{Adv}(M^D(q_0, \ldots, q_{m-1}, B, \boldsymbol{R}), f) \leqslant P_f \cdot 2^\epsilon \tag{5.3}$$

where P_f is the pattern observation probability associated with observer function f.

Proof of Theorem 2: We need to show that for every oracle replacing adversary:

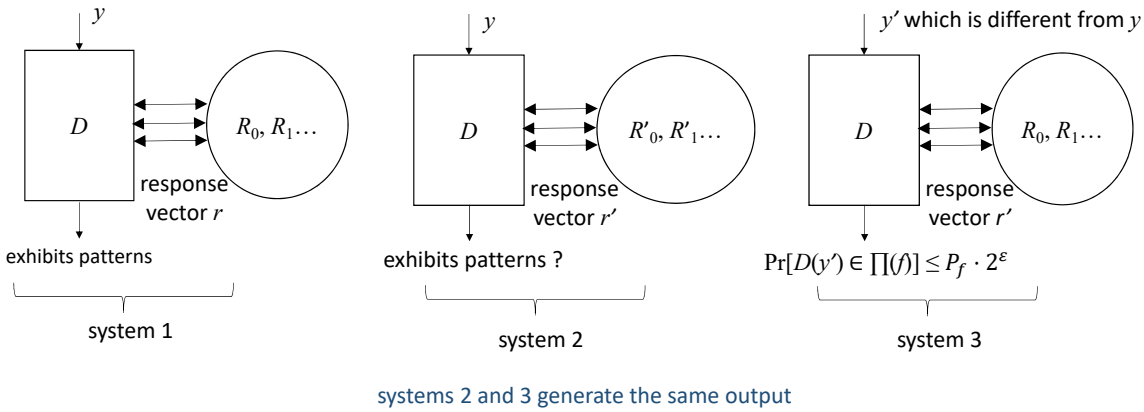


Figure 9: Replacing ingredient random permutations inside an RO^2 construction

$$\begin{gathered}\text{Prob}[\{\{R'_0, R'_1, \ldots\}, y\} \leftarrow M^D(q_0, \ldots, q_{m-1}, B, \boldsymbol{R});\ y \in \{q_0, \ldots, q_{m-1}\}; \\ \{R'_0, R'_1, \ldots\} \in \boldsymbol{R};\ D^{R'_0, R'_1, \cdots}(y) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^{\epsilon}\end{gathered} \tag{5.4}$$

We assume that an adversary exists for which the relation 5.4 does not hold. This adversary repeatedly succeeds in producing random permutation replacements and inputs y the outputs of which exhibit patterns with probability greater than $P_f \cdot 2^{\epsilon}$. We show that if such adversary exists then it is not possible for E, D to be $\mathbf{RO}^2(f, B, \epsilon)$ which contradicts our assumption. The proof is similar as in Theorem 1. We first state and prove a lemma that bounds the probability of seeing patterns in the output of an RO^2 construction once we replace the ingredient random permutations.

Lemma 1: Let's assume that we have a pair of encryption and decryption oracles $D, E \in \mathbf{RO}^2(f, B, \epsilon)$ for some f, B, ϵ, $D = E^{-1}$, and some input y, such that $D^{R_0, R_1, R_2, \cdots}(y) \in \mathbf{\Pi}(f)$. Then for any set of ingredient random permutation replacements $R'_0, R'_1, \ldots$ which are also random permutations, the probability $\text{Prob}[D^{R'_0, R'_1, \cdots}(y) \in \mathbf{\Pi}(f)]$ of seeing patterns in the output of y is bounded by:

$$\text{Prob}[D^{R'_0, R'_1, \cdots}(y) \in \mathbf{\Pi}(f) \mid D^{R_0, R_1, R_2, \cdots}(y) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^{\epsilon} \tag{5.5}$$

Proof of Lemma 1: We consider a system 1 shown in Figure 9, where the decryption oracle D accesses the original ingredient random permutations $R_0, R_1, R_2, \ldots$ and in this system one particular query to $R_0, R_1, R_2, \ldots$ returns a query response vector r. In another system, system 2, the original ingredient random permutations $R_0, R_1, R_2, \ldots$ are replaced by $R'_0, R'_1, \ldots$, and the same query now returns a different response vector r'. In system 3 of the figure, the decryption oracle D accesses the original ingredient random permutations $R_0, R_1, R_2, \ldots$, but in this system the corresponding query to $R_0, R_1, R_2, \ldots$ returns a response vector r', which is the same as the one returned by the permutations of system 2. As the ingredient random permutations $R_0, R_1, R_2, \ldots$ are bijective functions, they are invertible. By inverting the ingredient random permutations $R_0, R_1, R_2, \ldots$ on the response vector r', one computes an input y' which needs to be provided to system 3 in order for the ingredient random permutations of this system to return the same response vector r', which is returned in system 2. Due to $R_0, R_1, R_2, \ldots$ being bijective and the RO^2 construction constraints introduced in Figure 3, input y' must be different from y. Since $y' \neq y$, and system 3 is an RO^2 construction, then the output of system 3 exhibits patterns with probability:

$$\text{Prob}[D^{R_0,R_1,\cdots}(y') \in \mathbf{\Pi}(f) \mid D^{R_0,R_1,R_2,\cdots}(y) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^{\epsilon} \tag{5.6}$$

The proof of Lemma 1 completes by observing that systems 2 and 3 generate the same output. Hence:

$$\begin{gathered}\text{Prob}[D^{R'_0,R'_1,\cdots}(y) \in \mathbf{\Pi}(f) \mid D^{R_0,R_1,R_2,\cdots}(y) \in \mathbf{\Pi}(f)] = \\ \text{Prob}[D^{R_0,R_1,\cdots}(y') \in \mathbf{\Pi}(f) \mid D^{R_0,R_1,R_2,\cdots}(y) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^{\epsilon}\end{gathered} \tag{5.7}$$

and Lemma 1 is proven. We proceed with the proof of Theorem 2 by stating and proving one more Lemma:

Lemma 2: Let's consider an ensemble of ingredient random permutation sets $\{R_0^{(i)}, R_1^{(i)}, \ldots\}$, $i \geqslant 0$. Let's also consider constructions $D, E \in \mathsf{RO}^2(f, B, \epsilon)$ for some f, B, ϵ and $D = E^{-1}$. We further consider a set of input indices, $J_0 = 0$, $J_1 > J_0$, $J_2 > J_1, \ldots$ for which $J_{i+1} - J_i \leqslant B$ for all $i \geqslant 0$. Using the constructions E, D and the indices $J_0, J_1, \ldots$, we define permutation swapping constructions E', D' as constructions that accept as input discrete sequences of values $y_0, y_1, \ldots$, have infinite lifetime as opposed to bounded by B, and provide output which is obtained as follows:

$$E'(y_j) = E^{R_0^{(i)},R_1^{(i)},\cdots}(y_j) \quad \text{and} \quad D'(y_j) = D^{R_0^{(i)},R_1^{(i)},\cdots}(y_j) \quad \text{for all } y_j \text{ such that } J_i \leqslant j < J_{i+1} \tag{5.8}$$

If E', D' are defined by equation 5.8, then for any input sequence $\tilde{y}_0, \tilde{y}_1, \ldots$ to D' which is not repeating inside the index bounds defined by $J_0, J_1, \ldots$ the following inequality is true:

$$\text{Prob}[D'(\tilde{y}) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^{\epsilon} \tag{5.9}$$

where $\tilde{y}_{j'} \neq \tilde{y}_{j''}$ for all $J_i \leqslant j' < J_{i+1}$, $J_i \leqslant j'' < J_{i+1}$, $j' \neq j''$ and $i \geqslant 0$, and where the relation 5.9 holds for every input $\tilde{y} \in \{\tilde{y}_0, \tilde{y}_1, \ldots\}$.

Proof of Lemma 2: From the definition of equation 5.8 it is evident that E' and D' are also encryption and decryption oracles and that $D' = E'^{-1}$. The inputs to decryption oracle D' can either result in outputs with patterns or not. On the other hand, the inputs that result in patterns can be split into repeating inputs and non-repeating inputs, as inputs from the sequence $\tilde{y}_0, \tilde{y}_1, \ldots$ to D' may be repeating across index bounds. It is sufficient to show that the property $\text{Prob}[D'(\tilde{y}) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^{\epsilon}$ holds for the repeating inputs which produce at least one output with patterns. This is because the probability $\text{Prob}[D'(\tilde{y}) \in \mathbf{\Pi}(f)]$ is trivially 0 if it is known that inputs are always unsuccessful. On the other hand, for the non-repeating inputs the property does hold, as the constructions defined by E', D' are RO^2 inside the index bounds.

It is easy to see that, for the remaining case of repeating inputs that produce at least one output with patterns, the property $\text{Prob}[D'(\tilde{y}) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^{\epsilon}$ holds due to Lemma 1. Indeed let's consider some input $\tilde{y}_j$ such that $J_i \leqslant j < J_{i+1}$ for some $i \geqslant 0$. Let's also consider that this input, if passed to the decryption oracle D', produces output that exhibits patterns:

$$D'(\tilde{y}_j) = D^{R_0^{(i)},R_1^{(i)},\cdots}(\tilde{y}_j) \in \mathbf{\Pi}(f) \tag{5.10}$$

If this input $\tilde{y}_j$ appears in the input sequence again, outside the index bounds J_i and J_{i+1}, then the probability of seing patterns at the output of this repeated instance of $\tilde{y}_j$ is bounded

according to Lemma 1. Specifically, if value $\tilde{y}_j$ appears again inside the index bounds J'_i and $J_{i'+1}$ for some $i' \neq i$, then the output of the decryption oracle D' for this repeated instance is equal to $D^{R_0^{(i')},R_1^{(i')},\cdots}(\tilde{y}_j)$. If we apply Lemma 1 to the sets of ingredient random permutations $\{R_0^{(i)}, R_1^{(i)}, \ldots\}$ and $\{R_0^{(i')}, R_1^{(i')}, \ldots\}$ and to the input value $\tilde{y}_j$, we obtain inequality:

$$\text{Prob}[D^{R_0^{(i')},R_1^{(i')},\cdots}(\tilde{y}_j) \in \mathbf{\Pi}(f) \mid D^{R_0^{(i)},R_1^{(i)},\cdots}(\tilde{y}_j) \in \mathbf{\Pi}(f)] \leqslant P_f \cdot 2^{\epsilon} \tag{5.11}$$

which completes the proof of Lemma 2. We also note that the inequality 5.9 of Lemma 2 holds even if the event $D'(\tilde{y}) \in \mathbf{\Pi}(f)$ is conditioned upon inputs to D' which are different from $\tilde{y}$. This is due to the fact that the construction D' is RO^2 inside index bounds and the reasonable assumption that systems can be constructed in such a way that different input values do not affect instances of D that query different ingredient random permutations.

Completing the proof of Theorem 2: Any instance of D that queries ingredient random permutations from one of the sets of $\boldsymbol{R}$ is RO^2 by the definition of D and due to the fact that the permutations contained in the sets of $\boldsymbol{R}$ are random permutations. Similarly, any permutation swapping construction D' which is produced from D and has infinite lifetime, exhibits patterns in its output with probability which is bounded according to Lemma 2, provided that the input is non-repeating inside index bounds.

Now, let's suppose we have an oracle replacing adversary M, which is repeatedly successful with probability higher than the bound $P_f \cdot 2^{\epsilon}$ of relation 5.4. In every attack, this adversary succeeds in computing a different y value and a different set of ingredient random permutations from $\boldsymbol{R}$ all resulting in patterns with probability $> P_f \cdot 2^{\epsilon}$. Furthermore, as in the proof of Theorem 1, this adversary M can be turned into another adversary M' which always returns some output and is still successful in attacking D with probability $> P_f \cdot 2^{\epsilon}$. We consider that such attacks are repeated again and again. One can see that the attacks performed by adversary M' form a trace of queries to a permutation swapping construction D' as defined in Lemma 2. The expected value E of the ratio of successful queries to D' over all queries made needs to satisfy the inequality $E > P_f \cdot 2^{\epsilon}$, due to the assumption about the existence of the adversary M'. On the other hand, the same expected value needs to satisfy the inequality $E \leqslant P_f \cdot 2^{\epsilon}$ due to the fact that Lemma 2 holds, which is not possible. Hence, Theorem 2 is proven.

So far, we have been able to reduce the proofs which establish security in the input perturbing and oracle replacing adversary models to showing that the constructions used are RO^2 according to a chosen observer function, query budget and indistinguishability parameter. The implications from Theorems 1 and 2 is that no matter how the adversary corrupts the cipher text the probability that the patterns are visible in some plaintext can be bounded, if a construction is RO^2. In what follows we discuss an example of a 512-bit construction, called IVP, which is built from 128-bit block cipher building blocks and which is RO^2 according to the observer function $f_{4\times16}$.

6 Example of a cryptographic construction supporting implicit integrity

6.1 Construction overview

The cryptographic construction we discuss in this section, and which supports implicit integrity, called IVP, is shown in Figures 10 and 11. Figure 10 provides an overview of the encrypt and decrypt paths of the construction, while Figure 11 provides a description of the stages involved

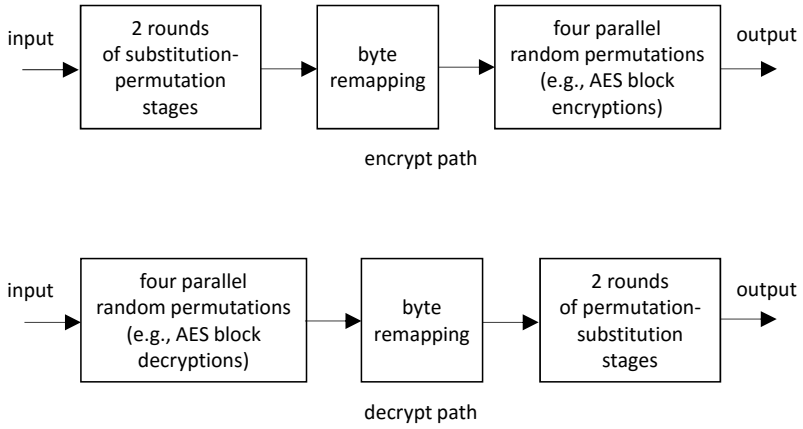


Figure 10: Overview of the encrypt and decrypt paths of the IVP construction

in the decrypt path. The IVP construction is a three level confusion diffusion network. It first employs two rounds of substitution and permutation stages followed by a byte remapping stage. The byte remapping stage prepares the inputs to four parallel random permutations, which provide an encryption result. On the decrypt path this order is reversed. The construction is 512-bit wide and its internal stages are defined for specific input and state lengths, which are discussed below.

The purpose of the two rounds of substitution and permutation stages is to diffuse every bit of the input into sets of 128 bits. Specifically, each bit is fully diffused into one of four sets of 128 bits. The purpose of the subsequent byte remapping stage is to change the order of bytes so that every 128-bit input, which is passed into the subsequent random permutations, contains bytes that depend on all 512 bits of the input. The random permutations of the construction can be realized using any block cipher which is 128-bit wide. For example, they can be realized as four independent AES block encryption stages. The preceding substitution and permutation stages can also be realized using AES round building blocks for convenience, as discussed later in this section.

The IVP construction, as we prove later, is indeed RO^2 for the $f_{4\times 16}$ pattern frequency observer, the life time value of $B = 2^{32}$ queries and the indistinguishability parameter $\epsilon = 2.651$ bits. For the sake of clarity, we remind that we are interested in the RO^2 behavior of the decrypt path of the construction, where the inputs are provided by an entity who has no knowledge of any keys used by the construction. So, the lifetime $B = 2^{32}$ is not the lifetime of a valid user of the construction but of an adversary (i.e., input perturbing or oracle replacing adversary) who attacks it.

6.2 On-line attacks and their implications

IVP supports implicit integrity at a security level of 32 bits. Such security level is lower than the security levels supported by standard MAC algorithms (e.g., [1] [2]), which are typically at 256 or 512 bits. Still, 32-bit security is not insignificant in the context of on-line attacks. In on-line attacks, the detection of even a single corruption exposes the attack and the adversary. As the adversary can only corrupt data in their encrypted form, any subsequent read operation performed on corrupted data results in a plaintext with high entropy with very high probability, thus exposing the attack. For this reason, even a lower level of security, such as at 32 bits, may be quite effective in protecting some computing systems. Defense against on-line attacks

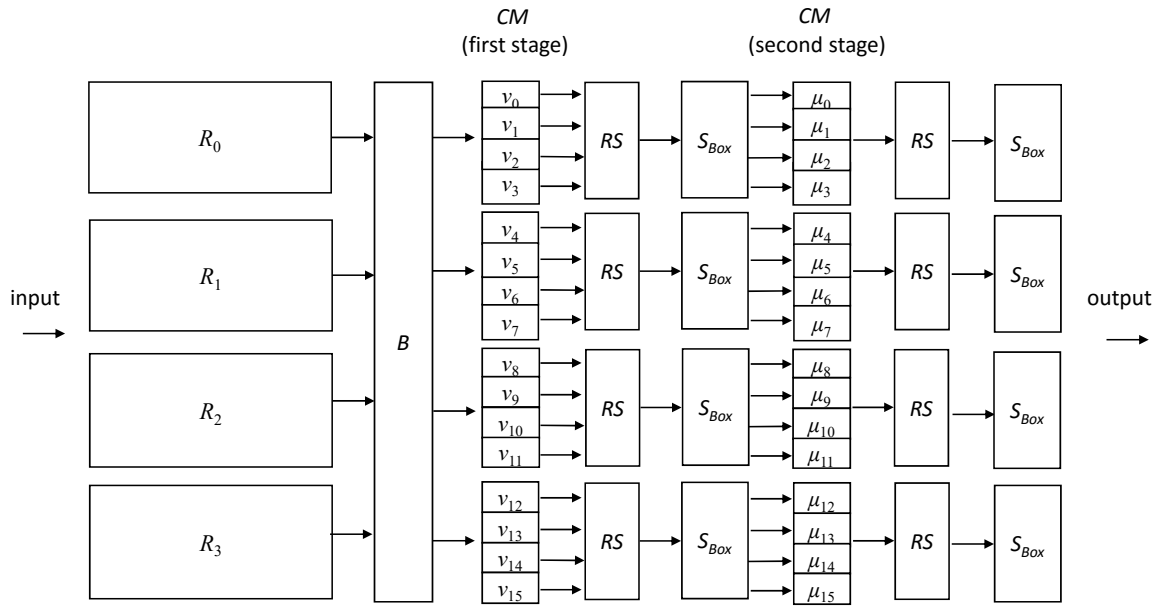


Figure 11: Description of the decrypt path of the IVP construction

at 32 bits of security also means that the probability of the adversary succeeding one and only time is $2^{-32+\epsilon}$ for some ϵ. Moreover, a single corruption detection can cause re-encryption of all user data with new keys, thus mitigating the attack. On-line attacks are discussed in RFC 4086 [27].

In the context of on-line attacks, a lower bound on the number of queries issued by an adversary can be set. Within such bound (e.g., 2^{32} queries) the block cipher stages employed safely approximate random functions. For example, if the output of an 128-bit random permutation is not truncated, then the advantage of distinguishing this random permutation from an 128-bit random function after 2^{32} queries are performed is still $< 2^{-20}$. Practically, this means that whereas bytes in the output of a random function are random and uniformly distributed, bytes in the output of a random permutation take every value from 0 to 255 with probability that differs from 2^{-8} by no more than a value $\Delta = 2^{-20}$. In the analysis that follows we will either consider the output bytes of the ingredient random permutations as uniformly distributed and statistically independent when the analysis is not impacted by the presence of a distinguishing advantage, or indeed associated with such advantage, which will be present in our proofs yet considered negligible. We will also be using the term 'almost' random, uniformly distributed and statistically independent to characterize bytes or words, when the statistical properties of bytes or words differ from the properties of random, uniformly distributed and statistically independent bytes or words by no more than $O(\Delta)$.

6.3 Construction stages

Figure 11 shows the decrypt path for the IVP construction. In this figure, the input is the cipher text and the output is the plaintext. In the figure, the four ingredient random permutations employed $R_0, \ldots, R_3$ are four symmetric encryption/decryption stages. Each stage applies on inputs of width W_1. In the specific design we propose $W_1 = 128$ bits and the four random permutations of the figure can be realized AES encryption/decryption stages. Encryption/decryption is repeated four times, each for a separate 128-bit block of the input.

Encryption/decryption uses a key value K, which is a vector of four concatenated encryption keys, one for each block, and, optionally, a tweak vector T.

The stage B indicates an entity reordering operation. Entities are groups of W_2 bits. In this design $W_2 = 8$ bits and this stage is a byte remapping operation. Entity reordering takes place across the entire width of the construction which is $4 \cdot W_1$ bits. Entity reordering is an interleaving operation that ensures that outputs of the four ingredient random permutations of the construction are evenly distributed among the subsequent processing stages. Such interleaving operation is further discussed below. Two subsequent stages denoted as 'CM' perform AES-like bit linear processing on their inputs, which we refer to as 'column mixing'. In one realization the CM stages may implement the inverse mix columns or the mix columns transformation of AES. In general, the requirement for each CM stage is to implement bit linear systems that connect m W_2-bit input entities to m W_2-bit output entities using an MDS matrix. The rank of each bit linear system for each output entity should be exactly W_2. It is easy to see that the AES mix columns and inverse mix columns transformations meet this requirement for $m = 4$ and $W_2 = 8$ bits. For the proofs discussed below we require that $m = 4$. In the figure there are 16 first stage CM transformations denoted as $\nu_0, \ldots, \nu_{15}$ and 16 second stage CM transformations denoted as $\mu_0, \ldots, \mu_{15}$.

The subsequent 'RS' stages indicate a 'Row Shifting' operation which is an entity reordering operation similar to B. Row shifting occurs only inside W_1 bit blocks and not across such blocks as in the case of B. RS stages operate on entities of W_2 bits and perform cyclic rotation of such entities by increasing the number of rotate positions one at a time row-by-row. As in AES, it is assumed that W_2 bit entities are arranged in a matrix formation. In this formation rows are cyclically rotated either to the left or to the right by a number of positions which is increasing by one row-by-row. Row 1 for instance may be shifted by one entity position to the left. Row 2 by two entity positions to the left etc. In one realization RS stages implement the inverse shift rows or the shift rows transformation of AES.

Sbox stands for substitution box. Substitution occurs in the granularity of W_2 bits as in the case of the CM and RS stages. Each Sbox stage in the figure is an array of multiple substitution boxes of width W_2 bits. A substitution box is a randomly chosen 8-bit Pseudo-Random Permutation (PRP) which can be realized in many ways, such as by combining key additions with strong non-linear bit mixing operations. In one realization the key values used by these PRPs are set at the beginning of the operation of the IVP construction. This is equivalent to selecting a set of W_2-bit PRPs at random in the beginning of operation and using these PRPs as substitution boxes. It is under these considerations that we have proven specific security claims for the IVP construction, which we discuss in this section. Since we can select PRPs once at the beginning of operation of the construction, this means that we can have a single key set for the Sbox stages, which is independent of the keys used by the block ciphers that implement the random permutations $R_0, \ldots, R_3$.

The entity reordering operation B of the IVP construction is further illustrated in Figure 12 for the case where $W_2 = 8$ bits (byte remapping). Here, each byte is reordered to a new position so that all bytes coming from the same 128-bit block output are evenly distributed to all 128-bit block inputs of a next stage. For instance, regarding the outputs coming from a first random permutation R_0, a first byte is mapped to a first position in a next block. A second byte is mapped to a fourth position. A third byte to an eighth position, and so on. Similarly, concerning the outputs coming from a second random permutation R_1, a first byte is mapped to a second position. A second byte is mapped to a fifth position. A third byte is mapped to a ninth position, and so on.

To derive our byte remapping scheme we considered all possible ways to place 8 bits coming from the output of an ingredient random permutation into a 32-bit entity. This number which

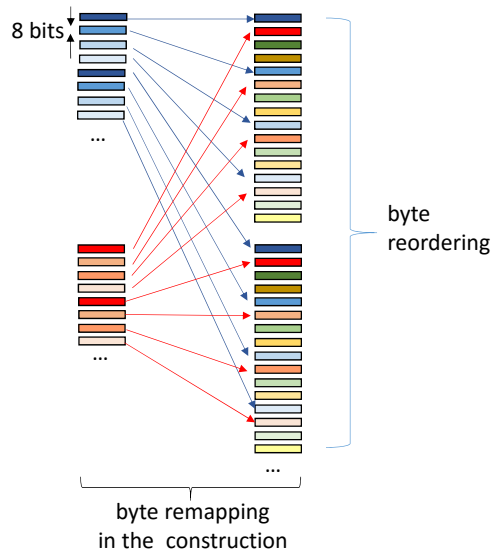


Figure 12: Byte remapping in the IVP construction

is equal to $\binom{32}{8} = 10,518,300$ is tractable and allows for the space to be searched even with exhaustive search. Each bit placement choice corresponds to a different bit linear system connecting the input bits of the *CM* transformation to the output bits. The bit placement choice determines which columns of the *CM* system matrix are selected in order to describe the input output relationship. As stated earlier, it is desirable for the rank of the system characterizing the derivation of each output byte to be exactly equal to $W_2 = 8$ bits. For the four output bytes we would like to have a cumulative rank equal to 32 bits. From the 10,518,300 choices 158,382 choices have so, including the choice of Figure 12.

6.4 Covered, uncovered and partially covered words

To prove that the IVP construction is RO^2 we first demonstrate that the flow of differentials characterizing this construction is associated with values that are almost random, uniformly distributed and statistically independent, where the term almost random, uniformly distributed and statistically independent is defined in Section 6.2. For this purpose we introduce the concept of covered, uncovered and partially covered words and bytes shown in Figure 13. A word w_0 in the output is covered, $w_0 \in C(\boldsymbol{W})$, if the differential p_0 which is superimposed on its state x_0 as a result of some input perturbation is (i) non-zero, (ii) almost random, uniformly distributed, and (iii) almost statistically independent from other word differentials. A word w_0 is uncovered $w_0 \in U(\boldsymbol{W})$, if the differential p_0 superimposed on its state x_0 as a result of some input perturbation is zero (e.g., the word may be exhibiting patterns).

Similarly, a byte b_0 in the output is covered, $b_0 \in C(\boldsymbol{B})$, if the differential p_0 which is superimposed on its state x_0 as a result of some input perturbation is (i) non-zero, (ii) almost random, uniformly distributed, and (iii) almsot statistically independent from other byte differentials. A byte b_0 is uncovered $b_0 \in U(\boldsymbol{B})$, if the differential p_0 superimposed on its state x_0 as a result of some input perturbation is zero. Finally, a word w_0 is partially covered $w_0 \in P(\boldsymbol{W})$ if the differential p_0 superimposed on its state x_0 causes one byte to be uncovered and the other byte to be covered.

It is not difficult to see that the covered and uncovered states are mutually exclusive for bytes and, consequently, each word can be in only one of the three states: covered, uncovered

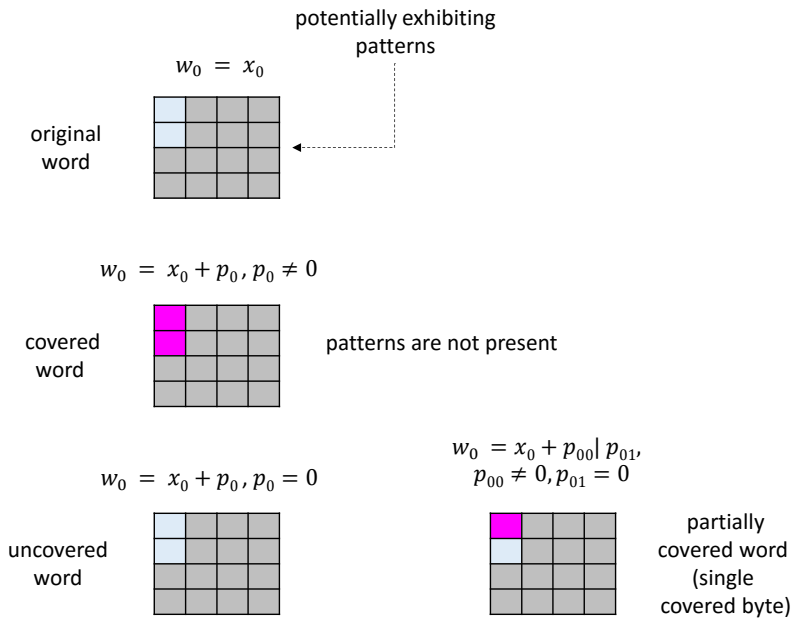


Figure 13: Covered, uncovered and partially covered words

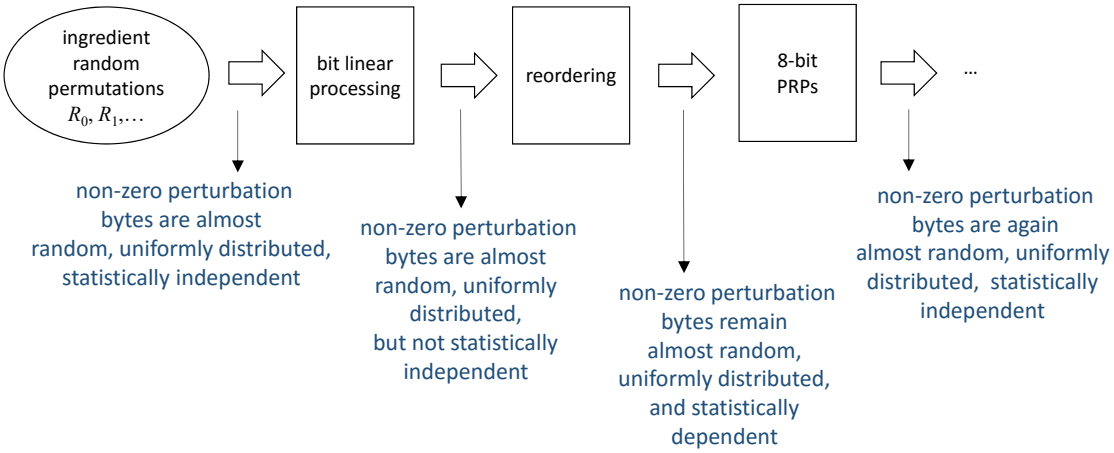


Figure 14: Statistical properties of differentials associated with ingredient random permutation outputs

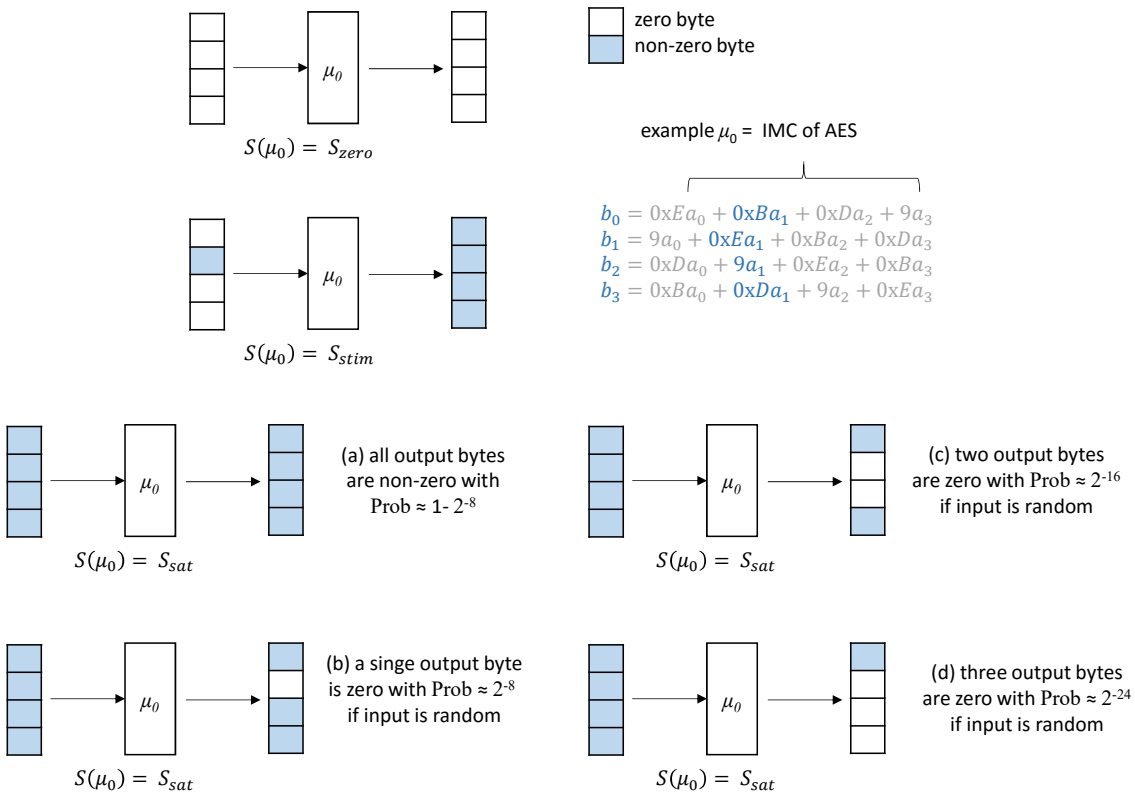


Figure 15: The states of a column mixing transformation

or partially covered. For the IVP construction, the mutual exclusivity of covered and uncovered states for bytes and can be established by observing that the ingredient random permutations are the sole source of non-zero byte differentials. Any subsequent processing involves only: (i) re-ordering operations at the byte granularity; (ii) full-rank bit-linear processing; and (iii) 8-bit PRPs. Based on this fact, byte differentials remain almost random, uniformly distributed and statistically independent as they flow through the IVP construction, as shown in Figure 14. First, at the output of the ingredient random permutation stages, non-zero byte differentials are almost random, uniformly distributed and statistically independent based on the indistinguishability of the ingredient random permutations from random functions and the life time limitation $B = 2^{32}$.

After the bit linear preprocessing stage, non-zero byte differentials are still almost random, uniformly distributed, but not necessarily statistically independent as each output byte is a linear combination of input bytes. After the byte remapping stage, non-zero byte differentials remain almost random, uniformly distributed, and possibly statistically dependent. It is in the last stage, and due to the fact that 8-bit PRPs are independently chosen at random, that non-zero byte differentials become again almost random, uniformly distributed and statistically independent. Furthermore, these properties of byte differentials do not change if additional diffusion stages or similar processing steps are added to the construction.

6.5 The state of column mixing transformations

The state $S(\mu_0)$ of a 4 byte column mixing transformation μ_0 can only be any of the following three of Figure 15: (i) zero state $S(\mu_0) = S_{zero}$. In this state all differential inputs $a_0, \ldots, a_3$ to

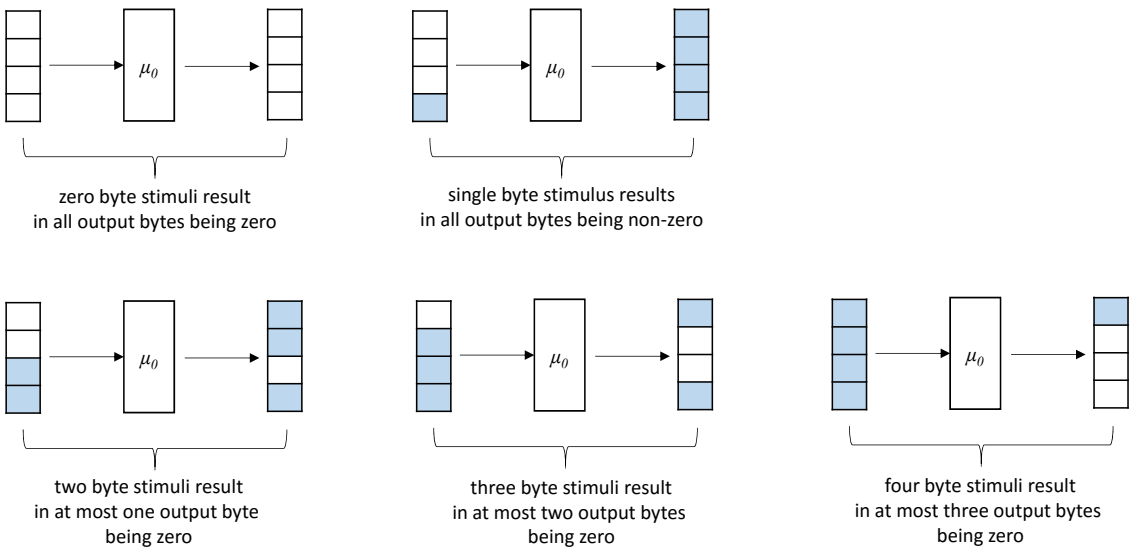


Figure 16: Behavior of a column mixing transformation as the number non-zero byte stimuli changes

the transformation μ_0 are equal to zero: $a_i = 0 \, \forall i \in [0, 3]$. Furthermore all output differentials are zero too; (ii) single byte stimulus state $S(\mu_0) = S_{stim}$. In this state only one of the input byte differentials is non-zero and all other byte differentials are zero: $\exists \, i \in [0, 3] : a_i \neq 0 \, \wedge \, (\forall j, j \neq i, j \in [0, 3], a_j = 0)$. Since the column mixing transformation matrix is MDS, all output byte differentials are non-zero; and (iii) saturation state $S(\mu_0) = S_{sat}$ where more than one of the input byte differentials is non-zero: $\exists \, q_0, \ldots, q_{m-1} \in [0, 3] : 1 < m \leqslant 4 \wedge a_{q_i} \neq 0 \forall i \in [0, m-1]$. Since the transformation μ_0 is bit linear of full rank, the following four possible events may be true: First, all output byte or byte differentials may be non-zero with probability $\approx 1 - 2^{-8}$. Second, a single output byte or byte differential may be zero with probability equal to $2^{-8} + O(\Delta)$ if inputs satisfy the statistical properties discussed above. Third, two output byte or byte differentials may be zero with probability $2^{-16} + 2^{-8} \cdot O(\Delta) + O(\Delta^2)$ if, again, inputs satisfy the statistical properties discussed above. Fourth, three output byte or byte differentials may be zero with probability $2^{-24} + 2^{-16} \cdot O(\Delta) + 2^{-8} \cdot O(\Delta^2) + O(\Delta^3)$ for the same inputs.

The effects different numbers of non-zero byte stimuli have on column mixing transformations are further illustrated in Figure 16. These observations are a direct consequence of the definition of the column mixing transformation being based on an MDS matrix. First, zero byte stimuli result in all output bytes being zero. Second, a single byte stimulus results in all output bytes being non-zero. Third, two byte stimuli result in at most one output byte being zero. Fourth, three byte stimuli result in at most two output bytes being zero. Finally, four byte stimuli result in at most three output bytes being zero.

6.6 On the security of the IVP construction

We begin the discussion on the security of the IVP construction by establishing the fact that the pattern frequency observer $f_{4\times16}$, which we choose to use is indeed associated with 32-bit security. This function observes whether there are 4 or more 16-bit words which are 16-bit aligned and equal to each other in a set of 512 bits.

Theorem 3: *About the security of the $f_{4\times16}$ pattern frequency observer.* The $f_{4\times16}$ pattern frequency observer is a PFO function associated with observation probability $2^{-32.866}$.

Proof of Theorem 3: We need to show that the probability of finding at least 4 16-bit words that are equal to each other in a set of 32 words in random data is $2^{-32.866}$. This is an instance of the more generic problem of computing the birthday collision probability $P^{(B)}(m, n, |V|)$ for a number of people $m > 1$ having the same birthday from among the members of a set n, where the number of birthdays is $|V|$. In this case $|V| = 65536$. For this problem solutions exist [13, 15, 14, 16, 17] such as the the Suzuki et. al. bounds [16] and the approximation by Kounavis et. al. [17]. Using the Suzuki bounds, the probability of the observer function $f_{4\times16}$ returning true upon receiving some 512-bit input is computed and found to be $P^{(B)} < 2^{-32.8659}$. Using the Kounavis approximation, the same probability is found to be $P^{(B)} \approx 2^{-32.8662}$. Both numbers are consistent and establish the security of the $f_{4\times16}$ observer function.

Theorem 4: *On the security of the* IVP *construction.* The IVP construction is in the set $\mathbf{RO}^2(f_{4\times16}, 2^{32}, \epsilon)$, specified by the observer function $f_{4\times16}$, the adversary query budget 2^{32}, and the indistinguishability parameter ϵ, which is equal to 2.651 bits:

$$\mathsf{IVP} \in \mathbf{RO}^2(f_{4\times16}, 2^{32}, 2.651) \tag{6.1}$$

Proof of Theorem 4: Let's consider y to be the input to the IVP construction. We need to show that $\text{Prob}[\mathsf{IVP}(y) \in \mathbf{\Pi}(f_{4\times16})] \leqslant P_{f_{4\times16}} \cdot 2^{\epsilon}$ where $\epsilon = 2.651$. Moreover, we need to show that the probability bound $P_{f_{4\times16}} \cdot 2^{\epsilon}$ remains the same even when the event $\mathsf{IVP}(y) \in \mathbf{\Pi}(f_{4\times16})$ is conditioned upon inputs other than y. We consider y to be the sum of a state vector z and a perturbation vector p: $y = z + p$. In one case, the patterns of the $f_{4\times16}$ observer function are present in the output of the state vector z and remain uncovered. In this case, the probability $\text{Prob}[\mathsf{IVP}(y) \in \mathbf{\Pi}(f_{4\times16})]$ is bounded by the probability that 4 words or more in the IVP construction output remain uncovered.

In another case, patterns of the $f_{4\times16}$ appear when all words of the output of the IVP construction are covered. In this case, the probability $\text{Prob}[\mathsf{IVP}(y) \in \mathbf{\Pi}(f_{4\times16})]$ is equal to $P_{f_{4\times16}}$ plus a negligible term associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. In between these two cases, a number of other subevents are possible, where patterns associated with the $f_{4\times16}$ observer function are formed from both uncovered words in the IVP construction output, the values of which depend on the state vector z, and covered words, the values of which depend on both the state z and the perturbation vector p.

Before analyzing each of these subevents, we compute the probability of having i-th word in the output of the IVP construction uncovered. We denote such probability as $P_{UW}(i) = \text{Prob}[w_i \in U(\boldsymbol{W})]$. In what follows we demonstrate that that $P_{UW}(i) = P_{UW} \leqslant 2^{-16} + O(\Delta')$ and is independent of the word index i. The term $O(\Delta')$ is much smaller than 2^{-16} and can be considered negligible. To prove this bound for $P_{UW}(i)$ we state and prove a number of useful Lemmas.

Lemma 3: *On computing the probability of a single uncovered word if only one block perturbation value is non-zero.* If a block perturbation value p_0 (i.e., a perturbation on one of the four 128-bit blocks of the IVP construction input) is non-zero and all other block perturbations p_1, p_2, and p_3 are equal to zero then: (i) all four first stage *CM* transformations $\nu_0, \ldots, \nu_3$ of the IVP construction are in the single byte stimulus state; (ii) all four second stage *CM* transformations $\mu_0, \ldots, \mu_3$ of the IVP construction are in the saturation state; and (iii) the probability of a single uncovered word is equal to $2^{-16} + O(\Delta')$, where $O(\Delta')$ is negligible:

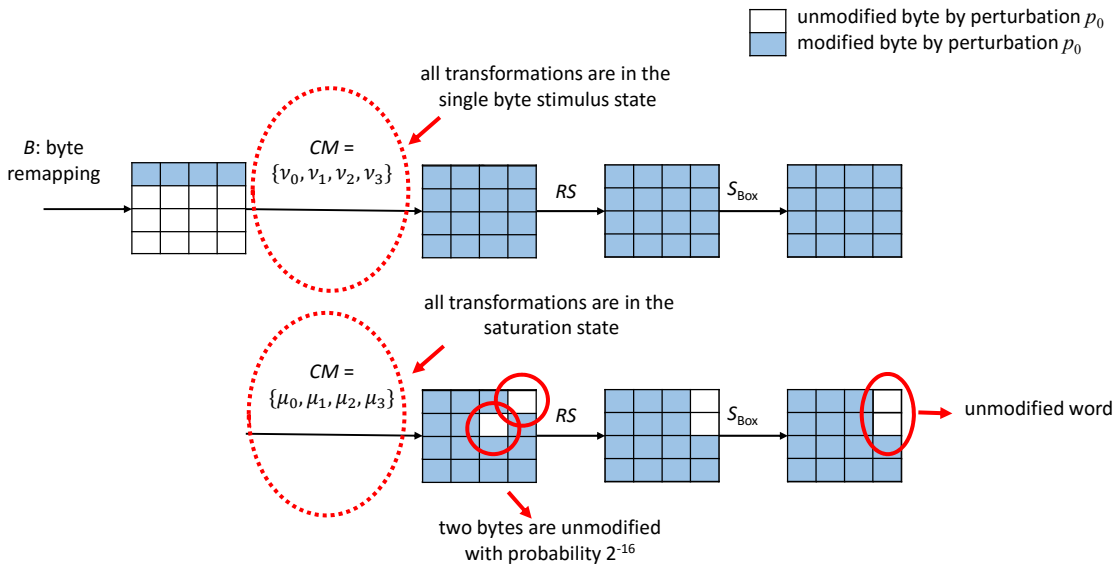


Figure 17: Corruptions in the IVP state when only one block perturbation is non-zero

$$\begin{gathered} \text{if } p_0 \neq 0 \text{ and } p_1 = p_2 = p_3 = 0, \text{ then } S(\nu_i) = S_{stim}\ \forall i \in [0,3], \\ S(\mu_i) = S_{sat}\ \forall i \in [0,3],\ P_{UW} = 2^{-16} + O(\Delta') \end{gathered} \tag{6.2}$$

Proof of Lemma 3: The situation where only one block perturbation is non-zero is shown in Figure 17. If only a single block perturbation is non-zero, then each column mixing transformation from among $\nu_0, \ldots, \nu_3$ has exactly one non-zero input byte differential. Hence, each transformation from $\nu_0, \ldots, \nu_3$ is in the single byte stimulus state. All four byte differentials of each column, after each transformation completes, are in this case non-zero. Byte differentials remain non-zero after the subsequent row shifting and *Sbox* stages complete too. Hence, all four second stage column transformations $\mu_0, \ldots, \mu_3$ of the IVP construction are in the saturation state. A single unmodified (i.e., uncovered) word in the output of the construction results from two unaligned unmodified bytes in the preceding row shifting transformation. Since each of the transformations $\mu_0, \ldots, \mu_3$ is in the saturation state each zero byte differential (i.e., unmodified byte) appears in the output of these transformations with probability $2^{-8} + O(\Delta)$. As a result the probability of a single uncovered word in the output of the IVP construction is equal to $2^{-16} + 2^{-8} \cdot O(\Delta) + O(\Delta^2)$. We set the negligible correcting term $2^{-8} \cdot O(\Delta) + O(\Delta^2)$ to $O(\Delta')$. This term compensates for the fact that the inputs are not exactly uniformly distributed, as they come from random permutations and not random functions. Hence, Lemma 3 is proven.

Lemma 4: *On computing the probability of a single uncovered word if exactly two block perturbation values are non-zero.* If two block perturbation values p_0 and p_1 are non-zero and the other perturbation values p_2 and p_3 are equal to zero then: (i) all four first stage *CM* transformations $\nu_0, \ldots, \nu_3$ of the IVP construction are in the saturation state; (ii) there can be at most a single second stage *CM* transformation in the zero state; and (iii) the probability of a single uncovered word is $\leqslant 2^{-16} + O(\Delta')$ for some negligible $O(\Delta')$.

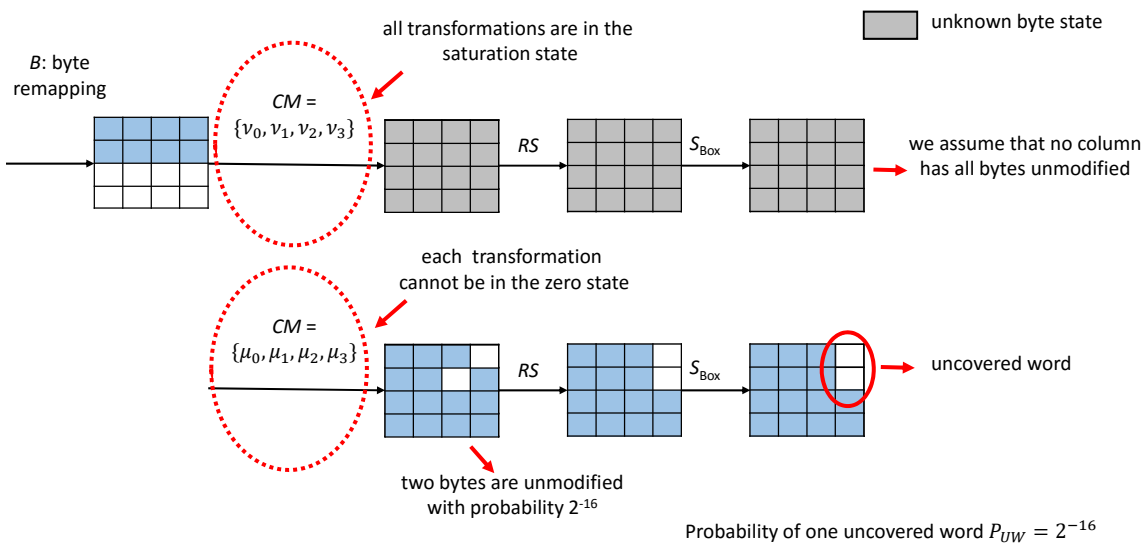


Figure 18: Corruptions in the IVP state when two block perturbations are non-zero (case I)

$$\begin{gathered}\text{if } p_0 \neq 0, p_1 \neq 0 \text{ and } p_2 = p_3 = 0 \text{ then } S(\nu_i) = S_{sat}\ \forall i \in [0,3],\\ \text{there is at most one transformation } \mu_q:\ S(\mu_q) = S_{zero},\ q \in [0,3],\\ P_{UW} \leqslant 2^{-16} + O(\Delta')\end{gathered} \tag{6.3}$$

Proof of Lemma 4: We consider two cases: In a first case (case I) we assume that no column has all bytes unmodified at the input to the second stage transformations $\mu_0, \ldots, \mu_3$. At another case (case II) we consider that this assumption does not hold. The situation of case I is shown in Figure 18. In case I, if exactly two block perturbations are non-zero then each column mixing transformation from $\nu_0, \ldots, \nu_3$ has exactly two non-zero input byte differentials. Hence, each transformation from $\nu_0, \ldots, \nu_3$ is in the saturation state. Furthermore, in each column output of $\nu_0, \ldots, \nu_3$ there can be at most one zero byte differential. Due to the assumption associated with case I, each transformation from $\mu_0, \ldots, \mu_3$ is either in the single byte stimulus or in the saturation state. Furthermore, due to the fact that there is at most one byte per column in the output of $\nu_0, \ldots, \nu_3$ which is unmodified, the number of transformations from $\mu_0, \ldots, \mu_3$ which are in the single byte stimulus state cannot be more than 1. Indeed, if there were 2 or more transformations from $\mu_0, \ldots, \mu_3$ in the single byte stimulus state, then the input to $\mu_0, \ldots, \mu_3$ would have contained at least 6 unmodified bytes. Therefore, either 3 or 4 transformations from $\mu_0, \ldots, \mu_3$ are in the saturation state. From these transformations two unaligned zero byte differentials result in a single uncovered word with probability $2^{-16} + 2^{-8} \cdot O(\Delta) + O(\Delta^2)$. Next, we set the correcting term $2^{-8} \cdot O(\Delta) + O(\Delta^2)$ to $O(\Delta')$, as in Lemma 3. In this way we complete the proof of Lemma 4 for case I.

If case II holds, then four unmodified bytes at the output of $\nu_0, \ldots, \nu_3$ become aligned via the subsequent row shifting transformation, in order to form a zero differential input column to one of the transformations $\mu_0, \ldots, \mu_3$. Since there can be at most four zero byte differentials at the output of $\nu_0, \ldots, \nu_3$ there can be no more than one unmodified input column to $\mu_0, \ldots, \mu_3$. Hence, exactly one transformation from $\mu_0, \ldots, \mu_3$ is in the zero state and three transformations are in the saturation state. Transformations $\nu_0, \ldots, \nu_3$ are all in the saturation state, as in

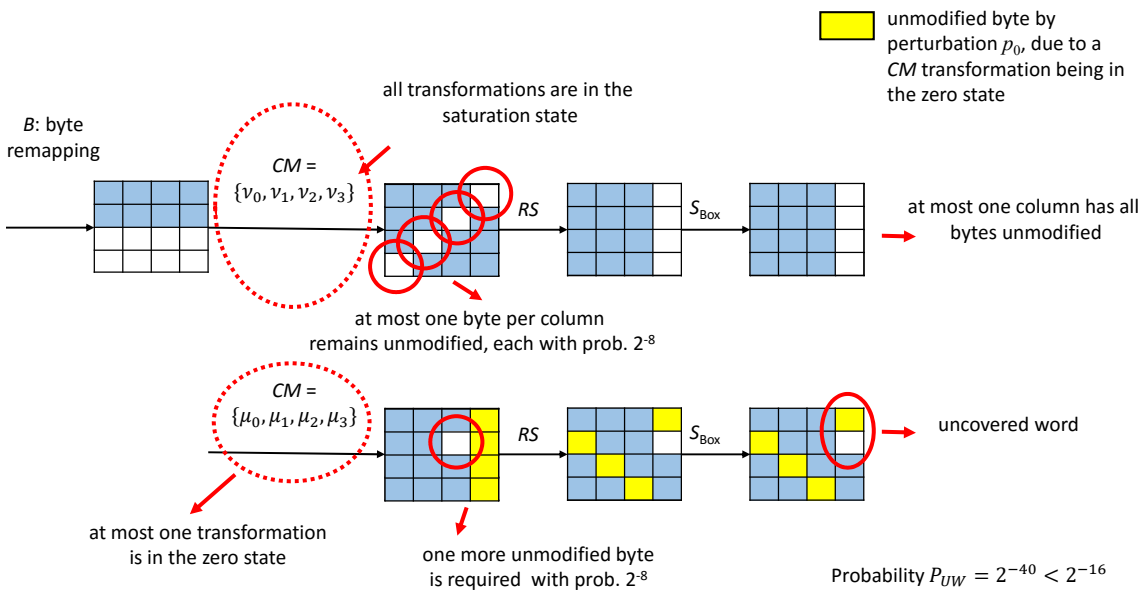


Figure 19: Corruptions in the IVP state when two block perturbations are non-zero (case II)

case I. Because of this reason, each of the four zero byte differentials at the output of $\nu_0, \ldots, \nu_3$ appears with probability $2^{-8} + O(\Delta)$. Having one word uncovered at the output of the IVP construction requires at least one more byte differential to be zero at the output of $\mu_0, \ldots, \mu_3$. As a result, the probability of having one uncovered word in the output of the IVP construction is $\leqslant 2^{-40} + 2^{-32} \cdot O(\Delta) + \ldots + O(\Delta^5) < 2^{-16} + O(\Delta')$, where $O(\Delta') = 2^{-32} \cdot O(\Delta) + \ldots + O(\Delta^5)$. Hence Lemma 4 is proven for case II as well.

Lemma 5: *On computing the probability of a single uncovered word if exactly three block perturbation values are non-zero.* If three block perturbation values p_0, p_1 and p_2 are non-zero and perturbation value p_3 is equal to zero then: (i) all four first stage *CM* transformations $\nu_0, \ldots, \nu_3$ of the IVP construction are in saturation state; (ii) there can be at most two second stage *CM* transformations in the zero state; and (iii) the probability of a single uncovered word is $\leqslant 2^{-16} + O(\Delta')$ for some negligible $O(\Delta')$:

$$\begin{gathered} \text{if } p_0 \neq 0, p_1 \neq 0, p_2 \neq 0 \text{ and } p_3 = 0 \text{ then } S(\nu_i) = S_{sat}\ \forall i \in [0,3], \\ \text{there are at most two transformations } \mu_q, \mu_r :\ S(\mu_q) = S(\mu_r) = S_{zero},\ q, r \in [0,3], \\ P_{UW} \leqslant 2^{-16} + O(\Delta') \end{gathered} \tag{6.4}$$

Proof of Lemma 5: We consider three cases: In a first case (case I) we assume that no column has all bytes unmodified at the input to the second stage transformations $\mu_0, \ldots, \mu_3$. At another case (case II) we consider that the number columns that have all bytes unmodified at the input to the second stage transformations $\mu_0, \ldots, \mu_3$ is two or more. A third case is when the number columns that have all bytes unmodified at the input to the second stage transformations $\mu_0, \ldots, \mu_3$ is exactly one. This third case is similar to case II of Lemma 4 and its proof is omitted.

The situation of case I is shown in Figure 20. In case I, if exactly three block perturbations are non-zero then each column mixing transformation from $\nu_0, \ldots, \nu_3$ has exactly three non-zero input byte differentials. Hence, each transformation from $\nu_0, \ldots, \nu_3$ is in the saturation

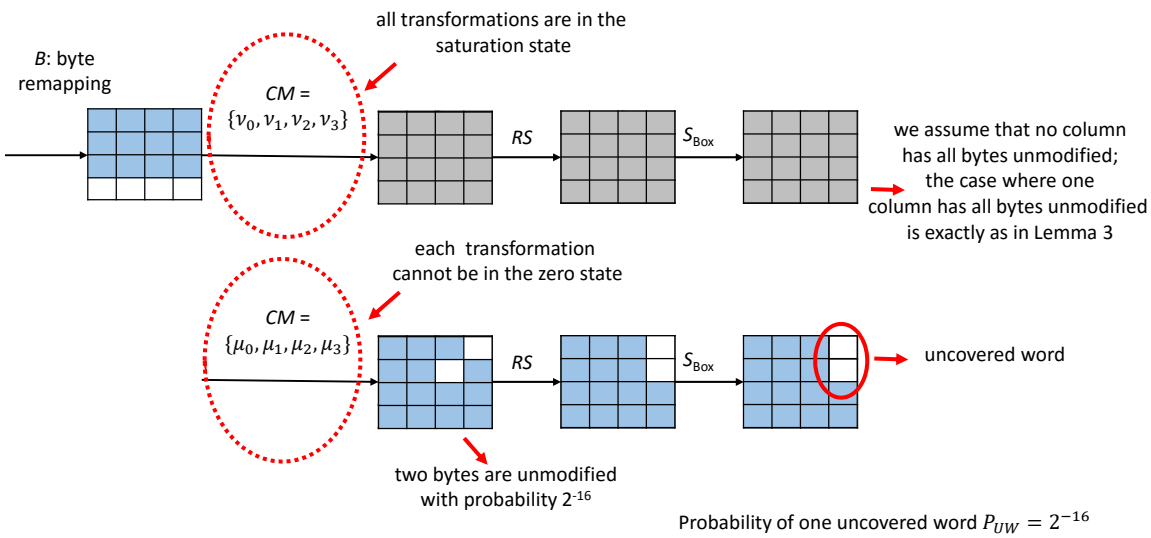


Figure 20: Corruptions in the IVP state when three block perturbations are non-zero (case I)

state. Furthermore, in each column output of $\nu_0, \ldots, \nu_3$ there can be at most two zero byte differentials. Due to the assumption associated with case I, each transformation from $\mu_0, \ldots, \mu_3$ is either in the single byte stimulus or in the saturation state. Furthermore, due to the fact that there are at most two byte differentials per column in the output of $\nu_0, \ldots, \nu_3$ which are zero, the number of transformations from $\mu_0, \ldots, \mu_3$ which are in the single byte stimulus state cannot be more than 2. If there were 3 or more transformations from $\mu_0, \ldots, \mu_3$ in the single byte stimulus state, then the input to $\mu_0, \ldots, \mu_3$ would have contained at least 9 unmodified bytes. Therefore 2, 3 or 4 transformations from $\mu_0, \ldots, \mu_3$ are in the saturation state. From these transformations, two unaligned byte differentials being equal to zero may result in a single uncovered word with probability $\leqslant 2^{-16} + O(\Delta')$, for some negligible $O(\Delta')$. Hence Lemma 5 is proven for case I.

If case II holds, then eight unmodified bytes at the output of $\nu_0, \ldots, \nu_3$ become aligned via the subsequent row shifting transformation, in order to form two zero differential input columns to two of the transformations $\mu_0, \ldots, \mu_3$. Since there can be at most eight zero byte differentials at the output of $\nu_0, \ldots, \nu_3$ there can be no more than two zero input columns to $\mu_0, \ldots, \mu_3$. Hence, exactly two transformations from $\mu_0, \ldots, \mu_3$ are in the zero state and two transformations are in the saturation state. Once again, transformations $\nu_0, \ldots, \nu_3$ are all in the saturation state, as in case I. Because of this reason, each of the eight zero byte differentials at the output of $\nu_0, \ldots, \nu_3$ appears with probability $2^{-8} + O(\Delta)$. Having two words uncovered at the output of the IVP construction is an event that results from having such unmodified byte differentials at the output of $\nu_0, \ldots, \nu_3$ as shown in the figure. As a result, the probability of having two uncovered words in the output of the IVP construction is $\leqslant 2^{-64} + 2^{-56} \cdot O(\Delta) + \ldots + O(\Delta^8) < (2^{-16})^2 + O(\Delta')$. As in Lemma 4, we set $2^{-56} \cdot O(\Delta) + \ldots + O(\Delta^8)$ to $O(\Delta')$. Hence Lemma 5 is proven for case II as well.

Lemma 6: *On computing the probability of a single uncovered word if all four block perturbation values are non-zero.* If all four block perturbation values p_0, p_1, p_2 and p_3 are non-zero then: (i) all four first stage *CM* transformations $\nu_0, \ldots, \nu_3$ of the IVP construction are in sat-

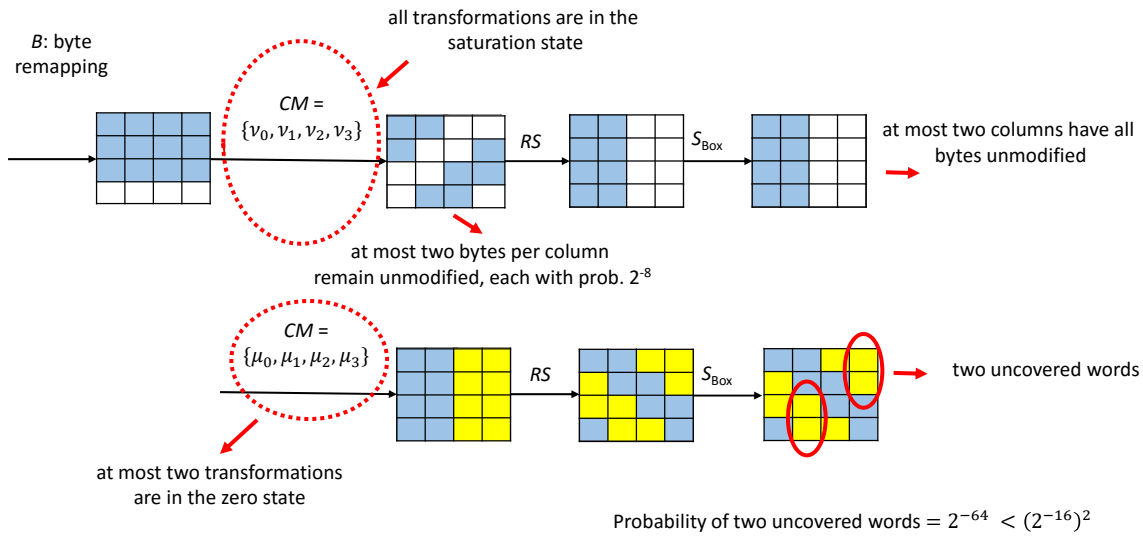


Figure 21: Corruptions in the IVP state when three block perturbations are non-zero (case II)

uration state; (ii) there can be at most three second stage *CM* transformations in the zero state; and (iii) the probability of a single uncovered word is $\leqslant 2^{-16} + O(\Delta')$ for some negligible $O(\Delta')$.

$$\begin{gathered} \text{if } p_i \neq 0,\ \forall i \in [0,3] \text{ then } S(\nu_i) = S_{sat}\ \forall i \in [0,3], \\ \text{there are at most three transformations } \mu_q, \mu_r, \mu_s : \\ S(\mu_q) = S(\mu_r) = S(\mu_s) = S_{zero},\ q,r,s \in [0,3],\ P_{UW} \leqslant 2^{-16} + O(\Delta') \end{gathered} \tag{6.5}$$

Proof of Lemma 6: We consider three cases: In a first case (case I) we assume that no column has all bytes unmodified at the input to the second stage transformations $\mu_0, \ldots, \mu_3$. At another case (case II) we consider that the number columns that have all bytes unmodified at the input to the second stage transformations $\mu_0, \ldots, \mu_3$ is three or more. A third case is when the number columns that have all bytes unmodified at the input to the second stage transformations $\mu_0, \ldots, \mu_3$ is either one or two. This third case is similar to the case II of Lemmas 4 and 5 and its proof is omitted.

The situation of case I is shown in Figure 22. In case I, if all four block perturbations are non-zero then each column mixing transformation from $\nu_0, \ldots, \nu_3$ has four non-zero input byte differentials. Hence, each transformation from $\nu_0, \ldots, \nu_3$ is in the saturation state in this case too. Furthermore, in each column output of $\nu_0, \ldots, \nu_3$ there can be at most three zero byte differentials. Due to the assumption associated with case I, each transformation from $\mu_0, \ldots, \mu_3$ is either in the single byte stimulus or in the saturation state. Furthermore, due to the fact that there are at most three byte differentials per column in the output of $\nu_0, \ldots, \nu_3$ which are zero, the number of transformations from $\mu_0, \ldots, \mu_3$ which are in the single byte stimulus state cannot be more than 3. If there were 4 transformations from $\mu_0, \ldots, \mu_3$ in the single byte stimulus state, then the input to $\mu_0, \ldots, \mu_3$ would have contained 12 unmodified bytes. Therefore at least one transformation from $\mu_0, \ldots, \mu_3$ is in the saturation state. From two such transformations in the saturation state, two unaligned byte differentials being equal

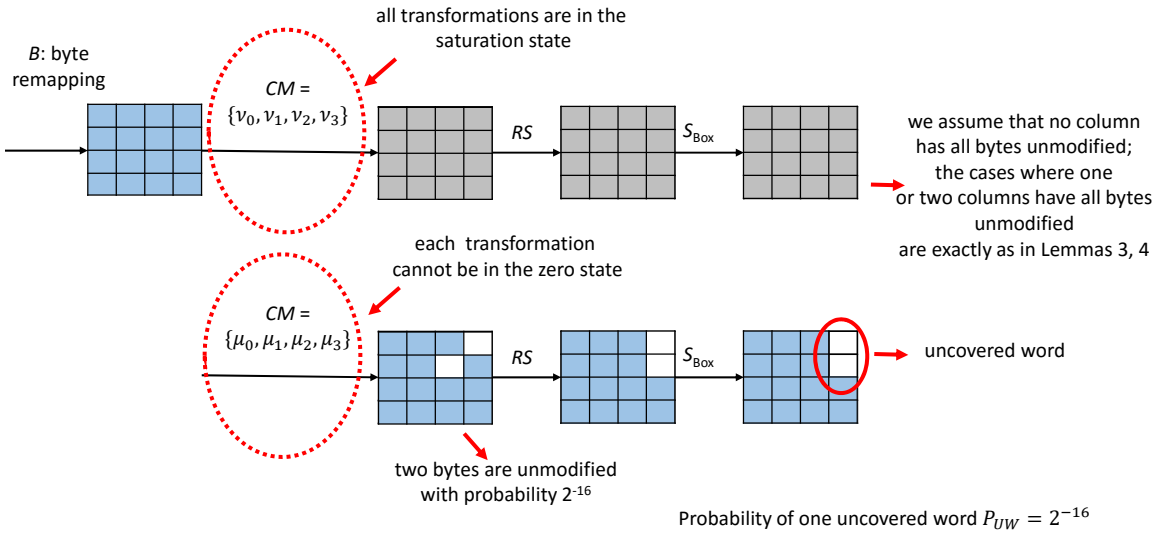


Figure 22: Corruptions in the IVP state when all block perturbations are non-zero (case I)

to zero result in a single uncovered word with probability $\leqslant 2^{-16} + O(\Delta')$, for some negligible $O(\Delta')$. Hence Lemma 6 is proven for case I.

If case II holds, then twelve unmodified bytes at the output of $\nu_0, \ldots, \nu_3$ become aligned via the subsequent row shifting transformation, in order to form three zero differential input columns to three of the transformations $\mu_0, \ldots, \mu_3$. Since there can be at most twelve zero byte differentials at the output of $\nu_0, \ldots, \nu_3$ there can be no more than three zero differential input columns to $\mu_0, \ldots, \mu_3$. Hence, exactly three transformations from $\mu_0, \ldots, \mu_3$ are in the zero state and one transformation is in the saturation state. Transformations $\nu_0, \ldots, \nu_3$ are all in the saturation state, as in case I. Because of this reason, each of the twelve zero byte differentials at the output of $\nu_0, \ldots, \nu_3$ appears with probability $2^{-8} + O(\Delta)$. Having four words uncovered at the output of the IVPconstruction is an event that results from having such unmodified bytes at the output of $\nu_0, \ldots, \nu_3$ as shown in the figure. As a result, the probability of having four uncovered words in the output of the IVP construction is equal to $\leqslant 2^{-96} + 2^{-88} \cdot O(\Delta) + \ldots + O(\Delta^{12}) < (2^{-16})^2 + O(\Delta')$. Again, we simplify this inequality setting $2^{-88} \cdot O(\Delta) + \ldots + O(\Delta^{12})$ to $O(\Delta')$. Hence Lemma 6 is proven for case II as well.

So far, Lemmas 3-6 have established the fact that the probability of having a single uncovered word appearing in the output of the IVP construction is bounded in a manner that is independent of the word index. Furthermore, the bound is equal to $2^{-16} + O(\Delta')$, where $O(\Delta')$ is a negligible term associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. A next corollary suggests a bound for the probability of having a single byte uncovered in the output of the IVP construction.

Corollary 1: A single uncovered byte appears at the output of the IVP construction with probability P_{UB} which is bounded as follows:

$$P_{UB} \leqslant 2^{-8} + O(\Delta) \tag{6.6}$$

where $Delta = 2^{-20}$ is associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions.

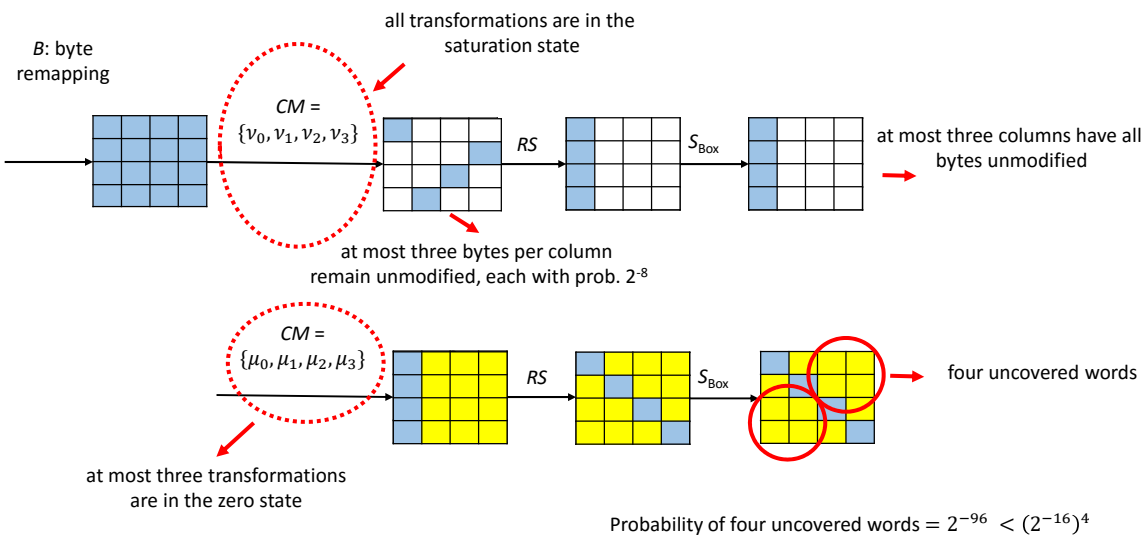


Figure 23: Corruptions in the IVP state when all block perturbations are non-zero (case II)

It is not difficult to see why Corollary 1 holds. In all situations covered by Lemma 3 and cases I and II of Lemmas 4, 5 and 6, either all of the column mixing transformations $\nu_0, \ldots, \nu_3$, or all of the column mixing transformations $\mu_0, \ldots, \mu_3$ are in the saturation state. Because of this fact, it is not possible for zero byte differentials to originate in any way other than from the bit linear mixing performed by these transformations. Hence, a byte is uncovered with probability $\leqslant 2^{-8} + O(\Delta)$.

Completing the proof of Theorem 4: Having established bounds for the probability of seeing an uncovered word and an uncovered byte at the output of the IVP construction, we proceed with proving Theorem 4. The probability of seeing the patterns associated with the $f_{4\times16}$ observer function at the output of IVP can be expressed as a sum of probabilities associated with different subevents. In one subevent, as discussed earlier, the patterns are present in the output of the state vector z (i.e., the original plaintext) and remain visible at the IVP output after the addition of a perturbation vector p onto z. In another subevent, patterns are formed only from word differentials, and all IVP output bytes and words are covered. In other subevents patterns are formed from both covered and uncovered words or bytes. All these subevents are captured in Proposition 1 below:

Proposition 1: The probability of the event that the output of the IVP construction exhibits the patterns associated with the $f_{4\times16}$ observer function is bounded as follows:

$$\text{Prob}[\mathsf{IVP}(y) \in \mathbf{\Pi}(f_{4\times16})] \leqslant P_{f_{4\times16}} + \binom{32}{4} \cdot P_{UW} + T + O(\Delta') \tag{6.7}$$

where $P_{f_{4\times16}}$ is the observation probability associated with $f_{4\times16}$, P_{UW} is the probability of a word in the output of IVP being uncovered, T is an additive term equal to $2^{-30.465}$ and $O(\Delta')$ can be considered negligible.

The term $P_{f_{4\times16}}$ corresponds to the subevent, where the pattern is formed only from word differentials and all words in the IVP output are covered. The term $\binom{32}{4} \cdot P_{UW}$ corresponds

to the subevent where the pattern is present in the output of the state vector z and remains visible after the addition of the perturbation vector p. In this case, the pattern is formed from words, all of which are uncovered. The additive term T corresponds to all other subevents where the pattern is formed from both covered and uncovered words or bytes. The proof of the correctness of Proposition 1 is provided in Appendix A.

Substituting $P_{f_{4\times16}}$ with $2^{-32.866}$ from Theorem 3, P_{UW} with the bound 2^{-16} from Lemmas 3-6, and T with $2^{-30.465}$ we obtain:

$$\text{Prob}[\mathsf{IVP}(y) \in \mathbf{\Pi}(f_{4\times16})] \leqslant 2^{-30.215} + O(\Delta) = 2^{-30.866} \cdot 2^{2.651} + O(\Delta') \tag{6.8}$$

To complete the proof we observe that the probability bounds of all sub-events considered in the derivation of 6.8 are independent of inputs other than y. Indeed all sub-events considered involve bytes or words which are either covered or demonstrate patterns coming from the unperturbed state z. The probability bounds associated with covered words or bytes are independent of the values of words or bytes of different inputs, as established by Lemmas 3-6 and Corollary 1. On the other hand, the bounds associated with sub-events where patterns may exist in the unperturbed state z are derived in a way that is independent of the values of these patterns. Therefore the bound in inequality 6.8 holds even if the event $\mathsf{IVP}(y) \in \mathbf{\Pi}(f_{4\times16})$ is conditioned upon inputs other than y. This completes the proof of Theorem 4 and establishes the security of the IVP construction in the input perturbing and oracle replacing adversary models.

7 Discussion

There are several questions about implicit integrity that are open and possibly the subject of future work. We believe that constructions which generalize IVP and apply to larger inputs may be able to support higher security levels. Such constructions would potentially relax the assumption of adversaries performing on-line attacks which characterizes the IVP example discussed in this paper. Specifically, the width values used in the IVP example can be replaced by generic width parameters and the pattern of seeing for our more words equal to each other in a set of 32 can be replaced by a more generic requirement that larger quantities (e.g, 128 bits, 256 bits) should demonstrate entropy below a threshold. Such generalization is the subject of future work.

One may also ask why not simply compress the data and augment it by a MAC in the now free space. We believe there is a practical reason why implicit integrity is better than compression. Compressing/decompressing in combinatorial logic requires not only detecting patterns, but also encoding the data in such a way so that some necessary space is freed for holding a MAC. For some patterns such as nibble-based patterns, this process can be quite costly, especially if implemented in combinatorial logic. Ongoing research of ours shows that the client cache lines that can be compressed at reasonable cost are significantly fewer than those protected via implicit integrity (78% as opposed to 91%). A MAC engine can also be costlier to implement in hardware. In contrast, the IVP construction requires only the detection of patterns, avoiding compressing or decompressing the data. Furthermore, it burdens the encrypt/decrypt data path with only two additional rounds of permutation-substitution steps, avoiding the use of any extra MAC engine in HW. Detailed comparison between compression-based and implicit integrity is the subject of future work.

References

[1] *Secure Hash Standard*, Federal Information Processing Standards Publication FIPS PUB 180-4.

[2] *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, Federal Information Processing Standards Publication FIPS PUB 202.

[3] *The Keyed-Hash Message Authentication Code (HMAC)*, Federal Information Processing Standards Publication FIPS PUB 198-1.

[4] *SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash*, NIST Special Publication 800-185.

[5] *Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication FIPS PUB 197.

[6] *Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices*, NIST Special Publication 800-38E.

[7] F. McKeen, I. Alexandrovich, A. Berenzon, C. Rozas, H. Shafi, V. Shanbhogue and U. Savagaonkar, *Innovative instructions and software model for isolated execution*, Proceedings of the Workshop on Hardware and Architectural Support for Security and Privacy (HASP), 2013.

[8] A. J. Menezes and P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[9] M. Bellare and P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*, In Proceedings, ACM Conference on Computer and Communications Security, pp. 62-73, 1993.

[10] M. Luby and C. Rackoff, *How to Construct Pseudorandom Permutations and Pseudorandom Functions*, SIAM Journal of Computing, Vol. 17, No, 2, 1988.

[11] C. Hall, D. A. Wagner, J. Kelsey and B. Schneier, *Building PRFs from PRPs*, CRYPTO 1998: 370-389.

[12] S. Gilboa and S. Gueron, *Distinguishing a truncated random permutation from a random function*, IACR Cryptology ePrint Archive 2015: 773 (2015).

[13] M. S. Klamkin and D. J. Newman, *Extensions on the Birthday Surprise*, Journal of Combinatorial Theory, Vol. 3, pp. 279-282, 1967.

[14] A. DasGupta, *The matching, birthday and the strong birthday problem: a contemporary review*, Journal of Statistical Planning and Inference, Vol. 130, pp. 377-389, 2004.

[15] *Wolfram Mathworld: Birthday Problem*, website, available on-line at: `http://mathworld.wolfram.com/BirthdayProblem.html`

[16] K. Suzuki, D. Tonien, K. Kurosawa and K. Toyota, *Birthday Paradox for Multi-collisions*, International Conference on Information Security and Cryptology, pp. 29-40, 2006.

[17] M. Kounavis, S. Deutsch, D. Durham and S. Komijani, *Non-recursive computation of the probability of more than two people having the same birthday*, ISCC 2017: 1263-1270.

[18] B. Sun, M, Liu, J. Guo, L. Qu and V. Rijmen, *New insights on AES-Like SPN Ciphers*, hskip 1em plus 0.5em minus 0.4emCRYPTO 2016.

[19] A. Bogdano and V. Rijmen, *Linear hulls with correlation zero and linear cryptanalysis of block ciphers*, Design, Codes and Cryptography, 70(3), pp. 369-383, 2014.

[20] R. Halevi, O. Goldreich and S. Halevi, *The random oracle metholodology revisited*, 30th ACM Symposium of the Theory of Computing (STOC), pp. 209-218, 1998.

[21] U. Maurer, R. Renner and C. Holenstein, *Indifferentability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology*, In Moni Naor, editor, First Theory of Cryptography Conference - TCC 2004, volume 2951 of LNCS, pages 21-39. Springer-Verlag, February 1921 2004.

[22] Y. Dodis, T. Liu, M. Stam, J. Steinberger, *Indifferentiability of Confusion-Diffusion Networks*, hskip 1em plus 0.5em minus 0.4emePrint 2015/680.

[23] I. Dinur, O. Dunkelman, N. Keller, A. Shamir, *Memory-Efficient Algorithms for Finding Needles in Haystacks*, CRYPTO 2016.

[24] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, NIST Special Publication 800-38D.

[25] J. Salowey, A. Choudhury and D. McGrew, *AES Galois Counter Mode (GCM) Cipher Suites for TLS*, RFC 5288.

[26] N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Calas and J. Walker, *The Skein Hash Function Family*, available online at `http://www.skein-hash.info/sites/default/files/skein1.1.pdf`

[27] *Randomness Requirements for Security*, RFC 4086, available online at `http://tools.ietf.org/html/rfc4086`. June 2006.

A Proof of the correctness of Proposition 1

For ease of notation, we denote the event $\mathsf{IVP}(y) \in \mathbf{\Pi}(f_{4\times16})$ as $E^{(f_{4\times16})}$. The probability of this event can be expressed as the sum of the probabilities of nine subevents, which are mutually exclusive:

$$
\begin{aligned}
\mathrm{Prob}[E^{(f_{4\times16})}] = \mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{no uncovered words}] \wedge [\text{no uncovered bytes}]\,] + \\
\mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{having exactly 1 uncovered word in the } \mathsf{IVP} \text{ output}]\,] + \\
\mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{having exactly 2 uncovered words in the } \mathsf{IVP} \text{ output}]\,] + \\
\mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{having exactly 3 uncovered words in the } \mathsf{IVP} \text{ output}]\,] + \\
\mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{having 4 or more uncovered words in the } \mathsf{IVP} \text{ output}]\,] + \\
\mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{no uncovered words}] \wedge [\text{exactly 1 uncovered byte in } \mathsf{IVP} \text{ output}]\,] + \\
\mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{no uncovered words}] \wedge [\text{exactly 2 uncovered bytes in } \mathsf{IVP} \text{ output}]\,] + \\
\mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{no uncovered words}] \wedge [\text{exactly 3 uncovered bytes in } \mathsf{IVP} \text{ output}]\,] + \\
\mathrm{Prob}[\,&[\text{seeing the } f_{4\times16} \text{ pattern in the } \mathsf{IVP} \text{ output}] \wedge \\
&[\text{no uncovered words}] \wedge [\text{4 or more uncovered bytes in } \mathsf{IVP} \text{ output}]\,]
\end{aligned}
\tag{A.1}
$$

To facilitate computations, we also introduce the following notation to refer to events and subevents associated with $E^{(f_{4\times16})}$:

i $E^{(f_{4\times16})}(N)$: seeing the $f_{4\times16}$ pattern in N out of 16 words of the IVP output.

ii $E^{(\text{ ex-uw})}(n, N)$: having exactly n uncovered words among N in the output of the IVP construction.

iii $E^{(\text{ at-uw})}(n, N)$: having at least n uncovered words among N in the output of the IVP construction.

iv $E^{(\text{ ex-eqw})}(n, N)$: having exactly one set of n equal covered words among N covered ones in the output of the IVP construction, where no other set with equal or greater number of equal covered words exists.

v $E^{(\text{ex-eqw})}(n, v, N)$: having exactly one set of n equal covered words among N covered ones in the output of the IVP construction, where no other set with equal or greater number of equal covered words exists, and where the value of these words is v.

vi $E^{(\text{at-eqw})}(n, N)$: having at least n equal covered words among N covered ones in the output of the IVP construction.

vii $E^{(\text{diff-uw})}(v)$: having an uncovered word in the output of the IVP construction with value different from v.

viii $E^{(\text{eq-uw})}(v)$: having an uncovered word in the output of the IVP construction with value equal to v.

ix $E^{(\text{ex-ub})}(n, N)$: having no uncovered words and exactly n uncovered bytes among N in the output of the IVP construction.

x $E^{(\text{at-ub})}(n, N)$: having no uncovered words and at least n uncovered bytes among N in the output of the IVP construction.

xi $E^{(\text{ex-eqb})}(n, N)$: having exactly one set of n equal covered bytes among N covered ones in the output of the IVP construction, where no other set with equal or greater number of equal covered bytes exists.

xii $E^{(\text{ex-eqb})}(n, v, N)$: having exactly one set of n equal covered bytes among N covered ones in the output of the IVP construction, where no other set with equal or greater number of equal covered bytes exists, and where the value of these bytes is v.

xiii $E^{(\text{eq-ub})}(v)$: having an uncovered byte in the output of the IVP construction with value equal to v.

xiv $E^{(\text{ex-uw-pat})}(n_0, n_1, N)$: having exactly n_0+n_1 uncovered words among N in the output of the IVP construction, of which n_0 words are elements of a set exhibiting the $f_{4\times16}$ pattern and the remaining n_1 are not.

xv $E^{(\text{at-uw-pat})}(n_0, n_1, N)$: having at least n_0+n_1 uncovered words among N in the output of the IVP construction, of which n_0 words are elements of a set exhibiting the $f_{4\times16}$ pattern and the remaining at least n_1 are not.

xvi $E^{(\text{ex-ub-pat})}(n_0, n_1, N)$: having exactly n_0+n_1 uncovered bytes among N in the output of the IVP construction, of which n_0 bytes are elements of a set exhibiting the $f_{4\times16}$ pattern and the remaining n_1 are not.

xvii $E^{(\text{at-ub-pat})}(n_0, n_1, N)$: having at least n_0+n_1 uncovered bytes among N in the output of the IVP construction, of which n_0 bytes are elements of a set exhibiting the $f_{4\times16}$ pattern and the remaining at least n_1 are not.

Using the notation above, we rewrite equation A.1 as follows:

$$
\begin{aligned}
\mathrm{Prob}[E^{(f_{4\times16})}] = \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{ex\text{-}ub})}(0,64)\,] + \\
\mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(1,32)\,] + \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(2,32)\,] + \\
\mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(3,32)\,] + \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{at\text{-}uw})}(4,32)\,] + \\
\mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{ex\text{-}ub})}(1,64)\,] + \\
\mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{ex\text{-}ub})}(2,64)\,] + \\
\mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{ex\text{-}ub})}(3,64)\,] + \\
\mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{at\text{-}ub})}(4,64)\,]
\end{aligned}
\tag{A.2}
$$

We further define probabilities $P_0, P_1, \ldots, P_8$ as shown in equation A.3:

$$
\begin{aligned}
P_0 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{ex\text{-}ub})}(0,64)\,] + \\
P_1 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(1,32)\,] \\
P_2 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(2,32)\,] \\
P_3 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(3,32)\,] \\
P_4 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{at\text{-}uw})}(4,32)\,] \\
P_5 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{ex\text{-}ub})}(1,64)\,] \\
P_6 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{ex\text{-}ub})}(2,64)\,] \\
P_7 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{ex\text{-}ub})}(3,64)\,] \\
P_8 &= \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(0,32) \ \wedge \ E^{(\ \mathsf{at\text{-}ub})}(4,64)\,]
\end{aligned}
\tag{A.3}
$$

so that:

$$
\mathrm{Prob}[E^{(f_{4\times16})}] = \sum_{i=0}^{8} P_i \tag{A.4}
$$

Lemma A.1 The probability P_1 defined in equation A.3 is bounded by $2^{-43.866} + \Delta_1$ where Δ_1 is a term significantly smaller than $2^{-43.866}$ and can thus be considered negligible:

$$
P_1 < 2^{-43.866} + \Delta_1 \tag{A.5}
$$

Proof of Lemma A.1 The probability P_1 can be expressed as the sum of the probabilities of two mutually exclusive subevents. The first subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there is exactly one uncovered word in this output and that the value of this uncovered word is not part of the visible pattern. The second subevent is the event that the $f_{4\times16}$ pattern is also visible in the output of the IVP construction, that there is exactly one uncovered word in the output, and that the value of this uncovered word is now part of the visible pattern:

$$
\begin{aligned}
P_1 = \mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(1,32) \ \wedge E^{(\ \mathsf{ex\text{-}uw\text{-}pat})}(0,1,32)\,] + \\
\mathrm{Prob}[E^{(f_{4\times16})}(32) \ \wedge \ E^{(\ \mathsf{ex\text{-}uw})}(1,32) \ \wedge E^{(\ \mathsf{ex\text{-}uw\text{-}pat})}(1,0,32)\,]
\end{aligned}
\tag{A.6}
$$

Equation A.6 can be rewritten as:

$$\begin{aligned} P_1 = \mathrm{Prob}[E^{(\text{ ex-uw})}(1,32)] \cdot \\ \mathrm{Prob}[E^{(f_{4\times 16})}(32) \ \wedge \ E^{(\text{ ex-uw-pat})}(0,1,32) \mid E^{(\text{ ex-uw})}(1,32)] + \\ \mathrm{Prob}[E^{(\text{ ex-uw})}(1,32)] \cdot \\ \mathrm{Prob}[E^{(f_{4\times 16})}(32) \ \wedge \ E^{(\text{ ex-uw-pat})}(1,0,32) \mid E^{(\text{ ex-uw})}(1,32)] \end{aligned} \tag{A.7}$$

Next, we observe that, if it is known that there is one uncovered word in the output of the IVP construction, then the event of having a visible pattern in the output and that the uncovered word is not part of the pattern is the same as the event that there are 4 or more words equal to each other among 31 in the output of IVP and that the uncovered word is not part of the pattern. Similarly, the event of having a visible pattern in the output and that the uncovered word is part of the pattern is the same as the event that there are 3 or more words equal to each other among 31 in the output of IVP and that that the uncovered word is part of the pattern:

$$\begin{aligned} P_1 = \mathrm{Prob}[E^{(\text{ ex-uw})}(1,32)] \cdot \\ \mathrm{Prob}[E^{(\text{ at-eqw})}(4,31) \ \wedge \ E^{(\text{ ex-uw-pat})}(0,1,32) \mid E^{(\text{ ex-uw})}(1,32)] + \\ \mathrm{Prob}[E^{(\text{ ex-uw})}(1,32)] \cdot \\ \mathrm{Prob}[E^{(\text{ at-eqw})}(3,31) \ \wedge \ E^{(\text{ ex-uw-pat})}(1,0,32) \mid E^{(\text{ ex-uw})}(1,32)] \end{aligned} \tag{A.8}$$

In the equality above, the probability of the event of having 4 or more words equal to each other among 31, which we denote as $E^{(\text{ at-eqw})}(4,31)$, can be safely approximated by the probability of the most dominant subevent of having exactly one set of 4 words equal to each other and no other sets of 4 or more word equalities, $E^{(\text{ ex-eqw})}(4,31)$, provided that we add some negligible term δ to the result. Indeed using the Suzuki bound $\mathrm{Prob}[E^{(\text{ at-eqw})}(4,31)] < 2^{33.0585}$, whereas the probability of the event $E^{(\text{ ex-eqw})}(4,31)$ is almost equal to $2^{33.0590}$. Their difference is $2^{-44.55}$ which can be considered negligible when compared to the the bound $2^{33.0585}$. This approximation continues to hold even when the event $E^{(\text{ at-eqw})}(4,31)$ is conditioned upon $E^{(\text{ ex-uw})}(1,32)$, due to the almost statistical independence of the two events, and provided that a correcting negligible term is present. The approximation also holds when we compute the probability of this event $E^{(\text{ at-eqw})}(4,31)$ together with the event that the value of the uncovered word participates (or not) in the visible pattern. The events $E^{(\text{ ex-uw-pat})}(0,1,32)$ and $E^{(\text{ ex-uw-pat})}(1,0,32)$ when considered in conjunction with $E^{(\text{ at-eqw})}(4,31)$ do not significantly alter the fact that $E^{(\text{ ex-eqw})}(4,31)$ is the dominant subevent. Finally, we note that a similar approximation holds for the event of having 3 or more words equal to each other among 31 words. Hence:

$$\begin{aligned} P_1 = \mathrm{Prob}[E^{(\text{ ex-uw})}(1,32)] \cdot \\ \mathrm{Prob}[E^{(\text{ ex-eqw})}(4,31) \ \wedge \ E^{(\text{ ex-uw-pat})}(0,1,32) \mid E^{(\text{ ex-uw})}(1,32)] + \\ \mathrm{Prob}[E^{(\text{ ex-uw})}(1,32)] \cdot \\ \mathrm{Prob}[E^{(\text{ ex-eqw})}(3,31) \ \wedge \ E^{(\text{ ex-uw-pat})}(1,0,32) \mid E^{(\text{ ex-uw})}(1,32)] \ + \ \delta \end{aligned} \tag{A.9}$$

In the next step of the proof, we express the events $E^{(\text{ex-eqw})}(4,31)$ and $E^{(\text{ex-eqw})}(3,31)$ as a union of mutually exclusive subevents $E^{(\text{ex-eqw})}(4,v,31)$ and $E^{(\text{ex-eqw})}(3,v,31)$, where in each subevent the elements of the sets of 3 or 4 words equal to each other take a specific value v.

$$\begin{aligned} P_1 = {} & \text{Prob}[E^{(\text{ex-uw})}(1,32)] \cdot \\ & \sum_v \text{Prob}[E^{(\text{ex-eqw})}(4,v,31) \ \wedge \ E^{(\text{ex-uw-pat})}(0,1,32) \mid E^{(\text{ex-uw})}(1,32)] + \\ & \text{Prob}[E^{(\text{ex-uw})}(1,32)] \cdot \\ & \sum_v \text{Prob}[E^{(\text{ex-eqw})}(3,v,31) \ \wedge \ E^{(\text{ex-uw-pat})}(1,0,32) \mid E^{(\text{ex-uw})}(1,32)] \ + \ \delta \end{aligned} \tag{A.10}$$

Equation A.10 can be further rewritten as:

$$\begin{aligned} P_1 = {} & \text{Prob}[E^{(\text{ex-uw})}(1,32)] \cdot \\ & \sum_v \text{Prob}[E^{(\text{ex-eqw})}(4,v,31) \ \wedge \ E^{(\text{diff-uw})}(v) \mid E^{(\text{ex-uw})}(1,32)] + \\ & \text{Prob}[E^{(\text{ex-uw})}(1,32)] \cdot \\ & \sum_v \text{Prob}[E^{(\text{ex-eqw})}(3,v,31) \ \wedge \ E^{(\text{eq-uw})}(v) \mid E^{(\text{ex-uw})}(1,32)] \ + \ \delta \\ = {} & \text{Prob}[E^{(\text{ex-uw})}(1,32)] \cdot \\ & \sum_v \big(\text{Prob}[E^{(\text{ex-eqw})}(4,v,31) \ \wedge \ E^{(\text{diff-uw})}(v) \mid E^{(\text{ex-uw})}(1,32)] + \\ & \text{Prob}[E^{(\text{ex-eqw})}(3,v,31) \ \wedge \ E^{(\text{eq-uw})}(v) \mid E^{(\text{ex-uw})}(1,32)]\big) \ + \ \delta \end{aligned} \tag{A.11}$$

The state of each word in the output of the IVP construction, as being covered or uncovered, is almost statistically independent from the state of other words. Because of this, the condition $E^{(\text{ex-uw})}(1,32)$ can be removed from the probabilities of events that describe equality of covered word values $\text{Prob}[E^{(\text{ex-eqw})}(4,v,31)]$ and $\text{Prob}[E^{(\text{ex-eqw})}(3,v,31)]$ in equation A.11. This, again, can be done provided that a correcting negligible term is present. Moreover, each probability term can be expressed as a product of two:

$$\begin{aligned} P_1 = {} & \text{Prob}[E^{(\text{ex-uw})}(1,32)] \cdot \\ & \sum_v \big(\text{Prob}[E^{(\text{ex-eqw})}(4,v,31)] \ \cdot \ \text{Prob}[E^{(\text{diff-uw})}(v) \mid E^{(\text{ex-uw})}(1,32)] + \\ & \text{Prob}[E^{(\text{ex-eqw})}(3,v,31)] \ \cdot \ \text{Prob}[E^{(\text{eq-uw})}(v) \mid E^{(\text{ex-uw})}(1,32)]\big) \ + \ \delta \end{aligned} \tag{A.12}$$

We conclude the proof observing that the probability terms $\text{Prob}[E^{(\text{ex-eqw})}(4,v,31)]$ and $\text{Prob}[E^{(\text{ex-eqw})}(3,v,31)]$ are the same for every value of v. Because of this reason, these probability terms can be taken out of the summation, where the term v can be replaced by some term v_0 that represents any value from 0 to 65536. We also observe that the probability

$\text{Prob}[E^{(\text{diff-uw})}(v) \mid E^{(\text{ex-uw})}(1,32)]$ can be replaced by 1 in order to obtain an upper bound and that $\sum_v \text{Prob}[E^{(\text{eq-uw})}(v) \mid E^{(\text{ex-uw})}(1,32)] = 1$.

As a result P_1 is bounded by:

$$\begin{aligned} P_1 < \text{Prob}[E^{(\text{ex-uw})}(1,32)] \cdot \\ \big(65536 \cdot \text{Prob}[E^{(\text{ex-eqw})}(4,v_0,31)] + \text{Prob}[E^{(\text{ex-eqw})}(3,v_0,31)]\big) \ + \ \delta \end{aligned} \tag{A.13}$$

The term $\text{Prob}[E^{(\text{ex-uw})}(1,32)]$ is the probability of having one uncovered word in the output of the IVP construction in any location, and is equal to $32 \cdot 2^{-16} + A_0$ where the term A_0 is negligible and is associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. On the other hand, the term $\text{Prob}[E^{(\text{ex-eqw})}(4,v_0,31)]$ is equal to $\binom{31}{4} \cdot 2^{-64} + A_1$, where the term A_1 is also negligible and is associated with the approximation error and the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. Finally, the term $\text{Prob}[E^{(\text{ex-eqw})}(3,v_0,31)]$ is equal to $\binom{31}{3} \cdot 2^{-48} + A_2$, where the term A_2 is negligible for too, the same reasons as A_1. From these equalities and relation A.13 we get:

$$P_1 < 2^{-43.866} + \Delta_1 \tag{A.14}$$

where Δ_1 is negligible and depends on A_0, A_1, A_2 and δ. In this way, Lemma A.1 is proven.

Lemma A.2 The probability P_2 defined in equation A.3 is bounded by $2^{-46.279} + \Delta_2$ where Δ_2 is a term significantly smaller than $2^{-46.279}$ and can thus be considered negligible:

$$P_2 < 2^{-46.279} + \Delta_2 \tag{A.15}$$

Proof of Lemma A.2 The probability P_2 can be bounded in a similar manner as probability P_1 in Lemma A.1. Probability P_2 can be expressed as the sum of the probabilities of three mutually exclusive subevents. The first subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are exactly two uncovered words in this output and that the values of these uncovered words are not part of the visible pattern. The second subevent is the event that the $f_{4\times16}$ pattern is also visible in the output of the IVP construction, that there are exactly two uncovered words in the output, and that the value of the first uncovered word is part of the visible pattern, but the value of the second is not. The third subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are exactly two uncovered words in the output, and that the values of both uncovered words are part of the visible pattern. It should be noted that the terms 'first' and 'second' here do not refer to any specific order of the uncovered words, but are merely used for differentiating between the two. The same applies to other similar subsequent uses of the terms first, second, etc. Considering only the dominant subevents, as in the proof of Lemma A.1, we write the probability P_2 as:

$$
\begin{aligned}
P_2 = \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(2,32)] \cdot \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,30)\ \wedge\ E^{(\ \mathsf{ex\text{-}uw\text{-}pat})}(0,2,32) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)] + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(2,32)] \cdot \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,30)\ \wedge\ E^{(\ \mathsf{ex\text{-}uw\text{-}pat})}(1,1,32) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)] + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(2,32)] \cdot \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,30)\ \wedge\ E^{(\ \mathsf{ex\text{-}uw\text{-}pat})}(2,0,32) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)]\ +\ \delta
\end{aligned}
\tag{A.16}
$$

Equation A.16 can be further rewritten, following a similar methodology as in the proof of Lemma A.1:

$$
\begin{aligned}
P_1 = \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(2,32)] \cdot \\
\sum_v \big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v,30)]\ \cdot\ \mathrm{Prob}[E^{(\ \mathsf{diff\text{-}uw})}(v) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)]^2 + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v,30)]\ \cdot\ 2 \cdot \mathrm{Prob}[E^{(\ \mathsf{diff\text{-}uw})}(v) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)] \cdot \\
\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)] + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v,30)]\ \cdot\ \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)]^2\big)\ +\ \delta
\end{aligned}
\tag{A.17}
$$

Next, we observe that the terms $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v,30)]$ and $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v,30)]$ are the same for every value of v and can be taken out of the summation, as in the proof of Lemma A.1. We also observe that the probability $\mathrm{Prob}[E^{(\ \mathsf{diff\text{-}uw})}(v) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)]$ can be replaced by 1 in order to obtain an upper bound for the probability P_2, and that $\sum_v \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)] = 1$. Finally, we observe that $\sum_v \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)]^2 < \sum_v \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v) \mid E^{(\ \mathsf{ex\text{-}uw})}(2,32)] = 1$. As a result P_2 is bounded by:

$$
\begin{aligned}
P_2 < \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(2,32)] \cdot \\
\big(65536 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0,30)\ +\ 2 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0,30) + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0,30)\big)\ +\ \delta
\end{aligned}
\tag{A.18}
$$

The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(2,32)]$ is the probability of having two uncovered words in the output of the IVP construction in any locations and is equal to $\binom{32}{2} \cdot 2^{-32} + A_0$ where the term A_0 is negligible and is associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. On the other hand, the term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0,30)]$ is equal to $\binom{30}{4} \cdot 2^{-64} + A_1$, where the term A_1 is also negligible and is associated with the approximation error and the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. Finally, the term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0,30)]$ is equal to $\binom{30}{3} \cdot 2^{-48} + A_2$, where the term A_2 is negligible for the same reasons, and the term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0,30)]$ is equal to $\binom{30}{2} \cdot 2^{-32} + A_3$, where the term A_3 is negligible too. From these equalities and relation A.18 we get:

$$
P_1 < 2^{-46.279} + \Delta_2 \tag{A.19}
$$

where Δ_2 is negligible and depends on A_0, A_1, A_2, A_3 and δ. In this way, Lemma A.2 is proven.

Lemma A.3 The probability P_3 defined in equation A.3 is bounded by $2^{-46.865} + \Delta_3$ where Δ_3 is a term significantly smaller than $2^{-46.865}$ and can thus be considered negligible:

$$P_3 < 2^{-46.865} + \Delta_3 \tag{A.20}$$

Proof of Lemma A.3 The probability P_3 can be expressed as the sum of the probabilities of four mutually exclusive subevents. The first subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are exactly three uncovered words in this output and that the values of these uncovered words are not part of the visible pattern. The second subevent is the event that the $f_{4\times16}$ pattern is also visible in the output of the IVP construction, that there are exactly three uncovered words in the output, and that the value of the first uncovered word is part of the visible pattern, but the values of the other two not. The third subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are exactly three uncovered words in the output, that the values of the first two uncovered words are part of the visible pattern, and that the value of the third uncovered word is not. A fourth subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are exactly three uncovered words in the output, and that the values of these uncovered words are all part of the visible pattern. Considering only the dominant subevents, the probability P_3 can be written as:

$$\begin{aligned}
P_3 = {} & \text{Prob}[E^{(\text{ex-uw})}(3,32)] \cdot \\
& \text{Prob}[E^{(\text{ex-eqw})}(4,29) \wedge E^{(\text{ex-uw-pat})}(0,3,32) \mid E^{(\text{ex-uw})}(3,32)] + \\
& \text{Prob}[E^{(\text{ex-uw})}(2,32)] \cdot \\
& \text{Prob}[E^{(\text{ex-eqw})}(3,29) \wedge E^{(\text{ex-uw-pat})}(1,2,32) \mid E^{(\text{ex-uw})}(3,32)] + \\
& \text{Prob}[E^{(\text{ex-uw})}(2,32)] \cdot \\
& \text{Prob}[E^{(\text{ex-eqw})}(2,29) \wedge E^{(\text{ex-uw-pat})}(2,1,32) \mid E^{(\text{ex-uw})}(3,32)] + \\
& \text{Prob}[E^{(\text{ex-uw})}(2,32)] \cdot \\
& \text{Prob}[E^{(\text{ex-uw-pat})}(3,0,32) \mid E^{(\text{ex-uw})}(3,32)]
\end{aligned} \tag{A.21}$$

Equation A.21 can be further rewritten as follows, following a similar methodology as in Lemma A.2:

$$
\begin{aligned}
P_3 = \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(3,32)] \cdot \\
\sum_v \big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v,29)] \ \cdot\ \mathrm{Prob}[E^{(\ \mathsf{diff\text{-}uw})}(v)\,|E^{(\ \mathsf{ex\text{-}uw})}(3,32)]^3 + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v,29)] \ \cdot\ 3 \cdot \mathrm{Prob}[E^{(\ \mathsf{diff\text{-}uw})}(v)\,|E^{(\ \mathsf{ex\text{-}uw})}(3,32)]^2 \cdot \\
\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v)\,|E^{(\ \mathsf{ex\text{-}uw})}(3,32)] + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v,29)] \ \cdot\ 3 \cdot \mathrm{Prob}[E^{(\ \mathsf{diff\text{-}uw})}(v)\,|E^{(\ \mathsf{ex\text{-}uw})}(3,32)] \cdot \\
\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v)\,|E^{(\ \mathsf{ex\text{-}uw})}(3,32)]^2 + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(1,v,29)] \ \cdot\ \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v)\,|E^{(\ \mathsf{ex\text{-}uw})}(3,32)]^3\big) \ +\ \delta
\end{aligned}
\tag{A.22}
$$

Next, we observe that all terms of the form $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(n,v,29)]$, $n = 1,2,3,4$ are the same for every value of v and can be taken out of the summation as in the previous two proofs, replacing v by v_0 to represent any value. We also observe that the probability $\mathrm{Prob}[E^{(\ \mathsf{diff\text{-}uw})}(v)|E^{(\ \mathsf{ex\text{-}uw})}(3,32)]$ can be replaced by 1 in order to obtain an upper bound for the probability P_3, and that $\sum_v \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v)\,|E^{(\ \mathsf{ex\text{-}uw})}(3,32)] = 1$. Finally, we observe that $\sum_v \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v)|\ E^{(\ \mathsf{ex\text{-}uw})}(3,32)]^3 < \sum_v \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}uw})}(v)|\ E^{(\ \mathsf{ex\text{-}uw})}(3,32)]^2$. As a result P_3 is bounded by:

$$
\begin{aligned}
P_3 < \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(3,32)] \cdot \\
\big(65536 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0,29)\ +\ 3 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0,29) + \\
3 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0,29) + \\
\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(1,v_0,29)\big) \ +\ \delta
\end{aligned}
\tag{A.23}
$$

The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}uw})}(3,32)]$ is the probability of having three uncovered words in the output of the IVP construction in any locations and is equal to $\binom{32}{3} \cdot 2^{-48} + A_0$ where the term A_0 is negligible and is associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0,29)]$ is equal to $\binom{29}{4} \cdot 2^{-64} + A_1$, where the term A_1 is also negligible and is associated with the approximation error and the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0,29)]$ is equal to $\binom{29}{3} \cdot 2^{-48} + A_2$, where the term A_2 is negligible for the same reasons. The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0,29)]$ is equal to $\binom{29}{2} \cdot 2^{-32} + A_3$, where the term A_3 is negligible too. Finally, the term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(1,v_0,29)]$ is equal to $29 \cdot 2^{-16} + A_4$, where the term A_4 is negligible. From these equalities and relation A.23 we get:

$$
P_3 < 2^{-46.865} + \Delta_3 \tag{A.24}
$$

where Δ_3 is negligible and depends on $A_0, \ldots, A_4$ and δ. This completes the proof of Lemma A.3.

Lemma A.4 The probability P_5 defined in equation A.3 is bounded by $2^{-34.406} + \Delta_5$ where Δ_5 is a term significantly smaller than $2^{-34.406}$ and can thus be considered negligible:

$$P_5 < 2^{-34.406} + \Delta_5 \tag{A.25}$$

Proof of Lemma A.4 The probability P_5 can be expressed as the sum of the probabilities of two mutually exclusive subevents. The first subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are no uncovered words but one uncovered byte in this output, and that this uncovered byte is not part of the visible pattern. The second subevent is the event that the $f_{4\times16}$ pattern is also visible in the output of the IVP construction, that there are no uncovered words but one uncovered byte in the output, and that the value of this uncovered byte is part of the visible pattern. According to such splitting, the probability P_5 can be written as:

$$\begin{aligned}
P_5 = {} & \text{Prob}[E^{(\text{ ex-uw})}(0,32) \wedge E^{(\text{ ex-ub})}(1,64)] \cdot \\
& \text{Prob}[E^{(f_{4\times16})}(32) \wedge E^{(\text{ ex-ub-pat})}(0,1,64) \mid E^{(\text{ ex-uw})}(0,32) \wedge E^{(\text{ ex-ub})}(1,64)] + \\
& \text{Prob}[E^{(\text{ ex-uw})}(0,32) \wedge E^{(\text{ ex-ub})}(1,64)] \cdot \\
& \text{Prob}[E^{(f_{4\times16})}(32) \wedge E^{(\text{ ex-ub-pat})}(1,0,64) \mid E^{(\text{ ex-uw})}(0,32) \wedge E^{(\text{ ex-ub})}(1,64)]
\end{aligned} \tag{A.26}$$

Next, we observe that $\text{Prob}[E^{(\text{ ex-uw})}(0,32) \wedge E^{(\text{ ex-ub})}(1,64)] \approx \text{Prob}[E^{(\text{ ex-ub})}(1,64)]$ and that the event $E^{(f_{4\times16})}(32)$ can be expressed in terms of the probabilities of 3 or 4 words or more being equal to each other, in the output of the IVP construction.

$$\begin{aligned}
P_5 = {} & \text{Prob}[E^{(\text{ ex-ub})}(1,64)] \cdot \\
& \text{Prob}[E^{(\text{ at-eqw})}(4,31) \wedge E^{(\text{ ex-ub-pat})}(0,1,64) \mid E^{(\text{ ex-ub})}(1,64)] + \\
& \text{Prob}[E^{(\text{ ex-ub})}(1,64)] \cdot \\
& \text{Prob}[E^{(\text{ at-eqw})}(3,31) \wedge E^{(\text{ ex-ub-pat})}(1,0,64) \mid E^{(\text{ ex-ub})}(1,64)] + \delta
\end{aligned} \tag{A.27}$$

where δ compensates for the approximation error and can be considered negligible.

Next, we approximate the probabilities of the events $E^{(\text{ at-eqw})}(4,31)$ and $E^{(\text{ at-eqw})}(3,31)$ with the probabilities of the most dominant subevents $E^{(\text{ ex-eqw})}(4,31)$ and $E^{(\text{ ex-eqw})}(3,31)$ as in the previous proofs. Furthermore, we denote the concatenation of two byte values v and w forming a word as $v|w$ and write:

$$\begin{aligned}
P_5 = {} & \text{Prob}[E^{(\text{ ex-ub})}(1,64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ ex-eqw})}(4,v|w,31) \wedge E^{(\text{ ex-ub-pat})}(0,1,64) \mid E^{(\text{ ex-ub})}(1,64)] + \\
& \text{Prob}[E^{(\text{ ex-ub})}(1,64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ ex-eqw})}(3,v|w,31) \wedge E^{(\text{ ex-ub-pat})}(1,0,64) \mid E^{(\text{ ex-ub})}(1,64)] + \delta
\end{aligned} \tag{A.28}$$

using a correcting term δ to compensate for the approximations.

We proceed with the proof observing that the event $E^{(\text{ex-ub-pat})}(0,1,64)$ of an uncovered byte not being part of a visible pattern can be removed in order to obtain an upper bound, being a subset of the entire sample space Ω. We also observe that the event $E^{(\text{ex-ub-pat})}(1,0,64)$ of an uncovered byte being part of a visible pattern is a subset of the event that an uncovered byte takes the value v or the value w, and its adjacent covered byte also takes the value v or the value w. Hence:

$$\begin{aligned} P_5 < \text{Prob}[E^{(\text{ex-ub})}(1,64)] \cdot \\ \sum_{v|w} \text{Prob}[E^{(\text{ex-eqw})}(4,v|w,31)|\, E^{(\text{ex-ub})}(1,64)] + \\ \text{Prob}[E^{(\text{ex-ub})}(1,64)] \cdot \\ \sum_{v|w} \text{Prob}[E^{(\text{ex-eqw})}(3,v|w,31) \wedge (E^{(\text{eq-ub})}(v) \vee E^{(\text{eq-ub})}(w)) \wedge \\ (E^{(\text{ex-eqb})}(1,v,1) \vee E^{(\text{ex-eqb})}(1,w,1))\, |E^{(\text{ex-ub})}(1,64)] + \delta \end{aligned} \tag{A.29}$$

Relation A.29 can be further rewritten as:

$$\begin{aligned} P_5 < \text{Prob}[E^{(\text{ex-ub})}(1,64)] \cdot \sum_{v|w} \big(\text{Prob}[E^{(\text{ex-eqw})}(4,v|w,31)] + \\ \text{Prob}[E^{(\text{ex-eqw})}(3,v|w,31)] \cdot \big(\text{Prob}[E^{(\text{eq-ub})}(v)\, |E^{(\text{ex-ub})}(1,64)] + \\ \text{Prob}[E^{(\text{eq-ub})}(w)\, |E^{(\text{ex-ub})}(1,64)]\big) \cdot \\ \big(\text{Prob}[E^{(\text{ex-eqb})}(1,v,1)] + \text{Prob}[E^{(\text{ex-eqb})}(1,w,1)]\big)\big) + \delta \end{aligned} \tag{A.30}$$

The terms $\text{Prob}[E^{(\text{ex-eqw})}(4,v|w,31)]$, $\text{Prob}[E^{(\text{ex-eqw})}(3,v|w,31)]$ and the probability sum $\big(\text{Prob}[E^{(\text{ex-eqb})}(1,v,1)] + \text{Prob}[E^{(\text{ex-eqb})}(1,w,1)]\big)$ are all independent of the exact values of v and w. These terms can be taken out of the summation, replacing v and w with v_0 and w_0 respectively in order to refer to any value of v, w. We also observe that $\sum_{v|w} \text{Prob}[E^{(\text{eq-ub})}(v) \mid E^{(\text{ex-ub})}(1,64)] = \sum_{v|w} \text{Prob}[E^{(\text{eq-ub})}(w) \mid E^{(\text{ex-ub})}(1,64)] = 256$. Based on these observations, the relation A.30 can be simplified as follows:

$$\begin{aligned} P_5 < \text{Prob}[E^{(\text{ex-ub})}(1,64)] \cdot \\ \big(65536 \cdot \text{Prob}[E^{(\text{ex-eqw})}(4,v_0|w_0,31) + 512 \cdot \text{Prob}[E^{(\text{ex-eqw})}(3,v_0|w_0,31) \cdot \\ \big(\text{Prob}[E^{(\text{ex-eqb})}(1,v_0,1)] + \text{Prob}[E^{(\text{ex-eqb})}(1,w_0,1)]\big)\big) + \delta \end{aligned} \tag{A.31}$$

The term $\text{Prob}[E^{(\text{ex-ub})}(1,64)]$ is the probability of having an uncovered byte in the output of the IVP construction in any location and is equal to $64 \cdot 2^{-8} + A_0$ where the term A_0 is negligible and is associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. The term $\text{Prob}[E^{(\text{ex-eqw})}(4,v_0|w_0,31)]$ is equal to $\binom{31}{4} \cdot 2^{-64} + A_1$, where the term A_1 is also negligible for the same reasons. The term

$\text{Prob}[E^{(\text{ex-eqw})}(3, v_0|w_0, 31)]$ is equal to $\binom{31}{3} \cdot 2^{-48} + A_2$, where the term A_2 is negligible. The term $\text{Prob}[E^{(\text{ex-eqb})}(1, v_0, 1)]$ is equal to $2^{-8} + A_3$, where the term A_3 is negligible. Finally, the term $\text{Prob}[E^{(\text{ex-eqb})}(1, w_0, 1)]$ is also equal to $2^{-8} + A_3$. From these equalities and relation A.31 we get:

$$p_5 < 2^{-34.406} + \Delta_5 \tag{A.32}$$

where Δ_5 is negligible and depends on $A_0, \ldots, A_3$ and δ. This completes the proof of Lemma A.4.

Lemma A.5 The probability P_6 defined in equation A.3 is bounded by $2^{-33.187} + \Delta_6$ where Δ_6 is a term significantly smaller than $2^{-33.187}$ and can thus be considered negligible:

$$P_6 < 2^{-33.187} + \Delta_6 \tag{A.33}$$

Proof of Lemma A.5 The probability P_6 can be expressed as the sum of the probabilities of three mutually exclusive subevents. The first subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are no uncovered words but two uncovered bytes in this output, and that no uncovered byte is part of the visible pattern. The second subevent is the event that the $f_{4\times16}$ pattern is also visible in the output of the IVP construction, that there are no uncovered words but two uncovered bytes in the output, and that the value of the first uncovered byte is part of the visible pattern, but the value of the second is not. The third subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are no uncovered words but two uncovered bytes in the output, and that the values of the uncovered bytes are part of the visible pattern. According to such splitting, and if include only the dominant subevents, the probability P_6 can be written as:

$$\begin{aligned}
P_6 = {} & \text{Prob}[E^{(\text{ex-ub})}(2, 64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ex-eqw})}(4, v|w, 30) \ \wedge \ E^{(\text{ex-ub-pat})}(0, 2, 64) \mid E^{(\text{ex-ub})}(2, 64)] + \\
& \text{Prob}[E^{(\text{ex-ub})}(2, 64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ex-eqw})}(3, v|w, 30) \ \wedge \ E^{(\text{ex-ub-pat})}(1, 1, 64) \mid E^{(\text{ex-ub})}(2, 64)] + \\
& \text{Prob}[E^{(\text{ex-ub})}(2, 64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ex-eqw})}(2, v|w, 30) \ \wedge \ E^{(\text{ex-ub-pat})}(2, 0, 64) \mid E^{(\text{ex-ub})}(2, 64)] \ + \ \delta
\end{aligned} \tag{A.34}$$

Next, we obtain a bound for the probability P_6, in a similar manner as in the proof of Lemma A.4, considering that the event of an uncovered byte being part of a visible pattern is a subset of the event that an uncovered byte takes the value v or the value w, and its adjacent covered byte also takes the value v or the value w:

$$
\begin{aligned}
P_6 < \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(2,64)] \ \cdot \ \sum_{v|w} \big(&\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v|w,30)] + \\
&2 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v|w,30)] \ \cdot \ \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\ |E^{(\ \mathsf{ex\text{-}ub})}(2,64)] + \\
&\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\ |E^{(\ \mathsf{ex\text{-}ub})}(2,64)]\big) \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big) + \\
&\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v|w,30)] \ \cdot \ \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\ |E^{(\ \mathsf{ex\text{-}ub})}(2,64)] + \\
&\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\ |E^{(\ \mathsf{ex\text{-}ub})}(2,64)]\big)^2 \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)^2\big) \ + \ \delta
\end{aligned}
\tag{A.35}
$$

The terms $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(n,v|w,30)]$, for $n = 2,3,4$, as well as the probability sum $\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] + \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)$ are all independent of the exact values of v and w. These terms can be taken out of the summation, as in the previous proof. We also observe that $\sum_{v|w} \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)|\ E^{(\ \mathsf{ex\text{-}ub})}(2,64)] + \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)|\ E^{(\ \mathsf{ex\text{-}ub})}(2,64)]\big)^2 < \sum_{v|w} \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\ |\ E^{(\ \mathsf{ex\text{-}ub})}(2,64)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\ |\ E^{(\ \mathsf{ex\text{-}ub})}(2,64)]\big) \ = \ 512$. Based on these observations, the relation A.35 can be simplified as follows:

$$
\begin{aligned}
P_6 < \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(2,64)] \cdot \\
&\big(65536 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0|w_0,30) + \\
&1024 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0|w_0,30) \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big) + \\
&512 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0|w_0,30) \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big)^2\big) \ + \ \delta
\end{aligned}
\tag{A.36}
$$

The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(2,64)]$ is the probability of having two uncovered bytes in the output of the IVP construction as part of any two different words and is equal to $4 \cdot \binom{32}{2} \cdot 2^{-16} + A_0$ where the term A_0 is negligible and is associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0|w_0,30)]$ is equal to $\binom{30}{4} \cdot 2^{-64} + A_1$, where the term A_1 is also negligible for the same reasons. The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0|w_0,30)]$ is equal to $\binom{30}{3} \cdot 2^{-48} + A_2$, where the term A_2 is negligible. Similarly, the term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0|w_0,30)]$ is equal to $\binom{30}{2} \cdot 2^{-32} + A_3$, where the term A_3 is negligible too. Finally, the term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)]$ is equal to $2^{-8} + A_4$ for some negligible A_4 and the term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]$ is also equal to $2^{-8} + A_4$. From these equalities and relation A.36 we get:

$$
p_6 < 2^{-33.187} + \Delta_6 \tag{A.37}
$$

where Δ_6 is negligible and depends on $A_0, \ldots, A_4$ and δ. In this way, Lemma A.5 is proven.

Lemma A.6 The probability P_7 defined in equation A.3 is bounded by $2^{-31.913} + \Delta_7$ where Δ_7 is a term significantly smaller than $2^{-31.913}$ and can thus be considered negligible:

$$P_7 < 2^{-31.913} + \Delta_7 \tag{A.38}$$

Proof of Lemma A.6 The probability P_7 can be expressed as the sum of the probabilities of four mutually exclusive subevents. The first subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are no uncovered words but three uncovered bytes in this output, and that no uncovered byte is part of the visible pattern. The second subevent is the event that the $f_{4\times16}$ pattern is also visible in the output of the IVP construction, that there are no uncovered words but three uncovered bytes in the output, and that the value of the first uncovered byte is part of the visible pattern, but the values of the other two are not. The third subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are no uncovered words but three uncovered bytes in the output, and that the values of the first two uncovered bytes are part of the visible pattern, but the value of the last is not. The last subevent is the event that the $f_{4\times16}$ pattern is visible in the output of the IVP construction, that there are no uncovered words but three uncovered bytes in the output, and that the values of all uncovered bytes are part of the visible pattern. Considering only the dominant subevents, the probability P_7 can be written as:

$$\begin{aligned}
P_7 = {} & \mathrm{Prob}[E^{(\text{ex-ub})}(3,64)] \cdot \\
& \sum_{v|w} \mathrm{Prob}[E^{(\text{ex-eqw})}(4,v|w,29) \ \wedge \ E^{(\text{ex-ub-pat})}(0,3,64) \mid E^{(\text{ex-ub})}(3,64)] + \\
& \mathrm{Prob}[E^{(\text{ex-ub})}(3,64)] \cdot \\
& \sum_{v|w} \mathrm{Prob}[E^{(\text{ex-eqw})}(3,v|w,29) \ \wedge \ E^{(\text{ex-ub-pat})}(1,2,64) \mid E^{(\text{ex-ub})}(3,64)] + \\
& \mathrm{Prob}[E^{(\text{ex-ub})}(3,64)] \cdot \\
& \sum_{v|w} \mathrm{Prob}[E^{(\text{ex-eqw})}(2,v|w,29) \ \wedge \ E^{(\text{ex-ub-pat})}(2,1,64) \mid E^{(\text{ex-ub})}(3,64)] + \\
& \mathrm{Prob}[E^{(\text{ex-ub})}(3,64)] \cdot \\
& \sum_{v|w} \mathrm{Prob}[E^{(\text{ex-ub-pat})}(3,0,64) \mid E^{(\text{ex-ub})}(3,64)] \ + \ \delta
\end{aligned} \tag{A.39}$$

Next, we obtain a bound for the probability P_7, in a similar manner as in the proof of Lemmas A.4 and A.5. We consider that the event of an uncovered byte being part of a visible pattern is a subset of the event that an uncovered byte takes the value v or the value w, and its adjacent covered byte also takes the value v or the value w:

$$
\begin{aligned}
P_7 < \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(3,64)] \ \cdot \ \sum_{v|w} \big(&\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v|w,29)] + \\
&3 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v|w,29)] \ \cdot \ \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\,|E^{(\ \mathsf{ex\text{-}ub})}(3,64)] + \\
&\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\,|E^{(\ \mathsf{ex\text{-}ub})}(3,64)]\big) \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big) + \\
&3 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v|w,29)] \ \cdot \ \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\,|E^{(\ \mathsf{ex\text{-}ub})}(3,64)] + \\
&\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\,|E^{(\ \mathsf{ex\text{-}ub})}(3,64)]\big)^2 \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)^2 + \\
&\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(1,v|w,29)] \ \cdot \ \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\,|E^{(\ \mathsf{ex\text{-}ub})}(3,64)] + \\
&\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\,|E^{(\ \mathsf{ex\text{-}ub})}(3,64)]\big)^3 \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)^3\big) \ + \ \delta
\end{aligned}
\tag{A.40}
$$

As in the proof of Lemma A.5, the terms $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(n,v|w,29)]$, for $n = 1,2,3,4$, as well as the probability sum $\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)$ are all independent of the exact values of v and w. These terms can be taken out of the summation, replacing v and w with v_0 and w_0 respectively in order to refer to any value for v, w. We also observe that $\sum_{v|w} \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v) \mid E^{(\ \mathsf{ex\text{-}ub})}(3,64)] + \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w) \mid E^{(\ \mathsf{ex\text{-}ub})}(3,64)]\big)^3 < \sum_{v|w} \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v) \mid E^{(\ \mathsf{ex\text{-}ub})}(3,64)] + \mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w) \mid E^{(\ \mathsf{ex\text{-}ub})}(3,64)]\big) = 512$. Based on these observations, we simplify the relation A.40 in the following way:

$$
\begin{aligned}
P_7 < \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(3,64)] \cdot \\
&\big(65536 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0|w_0,29) + \\
&1536 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0|w_0,29) \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big) + \\
&1536 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0|w_0,29) \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big)^2 + \\
&512 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(1,v_0|w_0,29) \cdot \\
&\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big)^3\big) \ + \ \delta
\end{aligned}
\tag{A.41}
$$

The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(3,64)]$ is the probability of having three uncovered bytes in the output of the IVP construction as part of any three different words and is equal to $8 \cdot \binom{32}{3} \cdot 2^{-24} \cdot (1-\frac{1}{8})^{29} + A_0$ where the term A_0 is negligible and is associated with the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0|w_0,29)]$ is equal to $\binom{29}{4} \cdot 2^{-64} + A_1$, where the term A_1 is also negligible for the same reasons. The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0|w_0,29)]$ is equal to $\binom{29}{3} \cdot 2^{-48} + A_2$, where the term A_2 is negligible. Similarly, the term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0|w_0,29)]$ is equal to $\binom{29}{2} \cdot 2^{-32} + A_3$, where the term A_3 is negligible too. The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(1,v_0|w_0,29)]$ is

equal to $29 \cdot 2^{-16} + A_4$, where the term A_4 is negligible. Finally, the term $\text{Prob}[E^{(\text{ex-eqb})}(1, v_0, 1)]$ is equal to $2^{-8} + A_5$ for some negligible A_5 and the term $\text{Prob}[E^{(\text{ex-eqb})}(1, w_0, 1)]$ is also equal to $2^{-8} + A_5$. From these equalities and relation A.41 we get:

$$p_7 < 2^{-31.913} + \Delta_7 \tag{A.42}$$

where Δ_7 is negligible and depends on $A_0, \ldots, A_5$ and δ. This completes the proof of Lemma A.6.

Lemma A.7 The probability P_8 defined in equation A.3 is bounded by $2^{-31.728} + \Delta_8$ where Δ_8 is a term significantly smaller than $2^{-31.728}$ and can thus be considered negligible:

$$P_8 < 2^{-31.728} + \Delta_8 \tag{A.43}$$

Proof of Lemma A.7 First, we observe that the probability P_8 is mostly determined by the most dominant subevent which is to have no uncovered words and exactly four uncovered bytes in the IVP construction output. This probability can be expressed as the sum of the probabilities of five mutually exclusive subevents. These are the subevents where the $f_{4\times16}$ pattern is visible in the output of the IVP construction, there are no uncovered words in the output, and the number of uncovered bytes that are part of the visible pattern ranges from 0 to 4. Considering only the dominant subevents of these subcases, the probability P_8 can be written in a similar manner as P_7:

$$\begin{aligned}
P_8 = {} & \text{Prob}[E^{(\text{ex-ub})}(4, 64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ex-eqw})}(4, v|w, 28) \ \wedge \ E^{(\text{ex-ub-pat})}(0, 4, 64) \mid E^{(\text{ex-ub})}(4, 64)] + \\
& \text{Prob}[E^{(\text{ex-ub})}(4, 64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ex-eqw})}(3, v|w, 28) \ \wedge \ E^{(\text{ex-ub-pat})}(1, 3, 64) \mid E^{(\text{ex-ub})}(4, 64)] + \\
& \text{Prob}[E^{(\text{ex-ub})}(4, 64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ex-eqw})}(2, v|w, 28) \ \wedge \ E^{(\text{ex-ub-pat})}(2, 2, 64) \mid E^{(\text{ex-ub})}(4, 64)] + \\
& \text{Prob}[E^{(\text{ex-ub})}(4, 64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ex-ub-pat})}(3, 1, 64) \mid E^{(\text{ex-ub})}(4, 64)] + \\
& \text{Prob}[E^{(\text{ex-ub})}(4, 64)] \cdot \\
& \sum_{v|w} \text{Prob}[E^{(\text{ex-ub-pat})}(4, 0, 64) \mid E^{(\text{ex-ub})}(4, 64)] \ + \ \delta
\end{aligned} \tag{A.44}$$

Next, we obtain a bound for the probability P_8, considering that the event of an uncovered byte being part of a visible pattern is a subset of the event that an uncovered byte takes the value v or the value w, and its adjacent covered byte also takes the value v or the value w:

$$
\begin{aligned}
P_8 < \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(4,64)] \ \cdot \ \sum_{v|w} \big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v|w,28)] + \\
4 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v|w,28)] \ \cdot \ \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\,|E^{(\ \mathsf{ex\text{-}ub})}(4,64)] + \\
\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\,|E^{(\ \mathsf{ex\text{-}ub})}(4,64)]\big) \cdot \\
\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big) + \\
6 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v|w,28)] \ \cdot \ \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\,|E^{(\ \mathsf{ex\text{-}ub})}(4,64)] + \\
\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\,|E^{(\ \mathsf{ex\text{-}ub})}(4,64)]\big)^2 \cdot \\
\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)^2 + \\
4 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(1,v|w,28)] \ \cdot \ \big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\,|E^{(\ \mathsf{ex\text{-}ub})}(4,64)] + \\
\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\,|E^{(\ \mathsf{ex\text{-}ub})}(4,64)]\big)^3 \cdot \\
\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)^3 + \\
\big(\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(v)\,|E^{(\ \mathsf{ex\text{-}ub})}(4,64)] + \\
\mathrm{Prob}[E^{(\ \mathsf{eq\text{-}ub})}(w)\,|E^{(\ \mathsf{ex\text{-}ub})}(4,64)]\big)^4 \cdot \\
\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)^4 \ \big) \ + \ \delta
\end{aligned}
\tag{A.45}
$$

The terms $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(n,v|w,28)]$, for $n = 1,2,3,4$, as well as the probability sum $\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w,1)]\big)$ are all independent of the exact values of v and w. These terms can be taken out of the summation, replacing v and w with v_0 and w_0 respectively in order to refer to any value for v, w. Based on these observations we can simplify the relation A.45 in the following way:

$$
\begin{aligned}
P_8 < \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(4,64)] \cdot \\
\big(65536 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(4,v_0|w_0,28) + \\
2048 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(3,v_0|w_0,28) \cdot \\
\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big) + \\
3072 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(2,v_0|w_0,28) \cdot \\
\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big)^2 + \\
2048 \cdot \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqw})}(1,v_0|w_0,28) \cdot \\
\big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big)^3 + \\
512 \cdot \big(\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,v_0,1)] \ + \ \mathrm{Prob}[E^{(\ \mathsf{ex\text{-}eqb})}(1,w_0,1)]\big)^4 \ \big) \ + \ \delta
\end{aligned}
\tag{A.46}
$$

The term $\mathrm{Prob}[E^{(\ \mathsf{ex\text{-}ub})}(4,64)]$ is the probability of having four uncovered bytes in the output of the IVP construction as part of any four different words and is equal to $16 \cdot \binom{32}{4} \cdot 2^{-32} \cdot (1 - \frac{1}{8})^{28} + A_0$ where the term A_0 is negligible and is associated with the advantage of distinguishing

the ingredient random permutations of the IVP construction from random functions. The term $\text{Prob}[E^{(\text{ex-eqw})}(4, v_0|w_0, 28)]$ is equal to $\binom{28}{4} \cdot 2^{-64} + A_1$, where the term A_1 is also negligible for the same reasons. The term $\text{Prob}[E^{(\text{ex-eqw})}(3, v_0|w_0, 28)]$ is equal to $\binom{28}{3} \cdot 2^{-48} + A_2$, where the term A_2 is negligible. Similarly, the term $\text{Prob}[E^{(\text{ex-eqw})}(2, v_0|w_0, 28)]$ is equal to $\binom{28}{2} \cdot 2^{-32} + A_3$, where the term A_3 is negligible too. The term $\text{Prob}[E^{(\text{ex-eqw})}(1, v_0|w_0, 28)]$ is equal to $28 \cdot 2^{-16} + A_4$, where the term A_4 is negligible. Finally, the term $\text{Prob}[E^{(\text{ex-eqb})}(1, v_0, 1)]$ is equal to $2^{-8} + A_5$ for some negligible A_5 and the term $\text{Prob}[E^{(\text{ex-eqb})}(1, w_0, 1)]$ is also equal to $2^{-8} + A_5$. From these equalities and relation A.46 we get:

$$p_8 < 2^{-31.728} + \Delta_8 \tag{A.47}$$

where Δ_8 is negligible and depends on $A_0, \ldots, A_5$ and δ. This completes the proof of Lemma A.7.

Lemma A.8 The probability P_0 defined in equation A.3 is bounded by $2^{-32.866} + \Delta_0$ where Δ_0 is a term significantly smaller than $2^{-32.866}$ and can thus be considered negligible:

$$P_0 < 2^{-32.866} + \Delta_0 \tag{A.48}$$

Proof of Lemma A.8 Probability P_0 is the probability of seeing the $f_{4\times16}$ pattern in the output of the IVP construction when all words of the output are covered. Lemma A.8 is mostly similar to Theorem 3. In what follows we repeat some of the findings from that proof for completeness. Probability P_0 is equal to the observation probability $P_{f_{4\times16}}$ of a PFO associated with this pattern, plus the advantage of distinguishing the ingredient random permutations of the IVP construction from random functions. The observation probability $P_{f_{4\times16}}$, on the other hand, is the birthday collision probability associated with seeing four or more equal words among 32 in the truncated output of a random oracle. Using the Suzuki bound we find that $P_{f_{4\times16}} < 2^{-32.866}$. Furthermore, by setting the distinguisher's advantage equal to a negligible Δ_0 the correctness of relation A.48 is shown and Lemma A.8 is proven.

Lemma A.9 The probability P_4 defined in equation A.3 is bounded by $2^{-48.866} + \Delta_4$ where Δ_4 is a term significantly smaller than $2^{-48.866}$ and can thus be considered negligible:

$$P_4 < 2^{-48.866} + \Delta_4 \tag{A.49}$$

Proof of Lemma A.9 Probability P_4 is the probability of seeing the $f_{4\times16}$ pattern in the output of the IVP construction when four or more words of the output are uncovered. This is bounded by the probability of having four or more words in the IVP output uncovered, which is further bounded by $\binom{32}{4} \cdot 2^{-64} + \Delta_4$ for some negligible Δ_4. This bound is further equal to $2^{-48.866} + \Delta_4$.

Completing the proof of the correctness of Proposition 1: Summing up the non-negligible terms of the probabilities P_1, P_2, P_3, P_5, P_6, P_7, and P_8 we compute a term T as follows:

$$T = \sum_{i \in \{1,2,3,5,6,7,8\}} [\,\text{non-negligible term of}\,]\, P_i = 2^{-30.465} \tag{A.50}$$

This is the additive term that appears in Proposition 1. On the other hand, the observation probability $P_{f_{4\times16}}$, which also appears in Proposition 1, is the non-negligible term of probability

P_0. Furthermore, the remaining non-negligible term $\binom{32}{4} \cdot P_{UW}$ in Proposition 1 is the non-negligible term of probability P_4. Finally, we set $O(\Delta') = \Delta_0 + \Delta_1 + \ldots + \Delta_8$. This concludes the proof.