Security Proof of JAMBU under Nonce Respecting and Nonce Misuse Cases

Geng Wang[⋆], Haiyang Zhang and Fengmei Liu

Science and Technology on Information Assurance Laboratory, Beijing, 100072, P.R. China

Abstract. JAMBU is an AEAD mode of operation which entered the third round of CAESAR competition. However, it does not have a security proof like other modes of operation do, and there was a cryptanalysis result that has overthrown the security claim under nonce misuse case by the designers. In this paper, we complement the shortage of the scheme by giving security proofs of JAMBU both under nonce respecting case and nonce misuse case. We prove that JAMBU under nonce respecting case has a slightly lower security than the birthday bound of n bits, and JAMBU under nonce misuse case has a tight security bound of $n/2$ bits.

Keywords: JAMBU, CAESAR Competition, Provable Security, Nonce-Misuse Resistance

1 Introduction

Authenticated encryption or usually known as authenticated encryption with associated data (AE or AEAD for short), which was formalized in [7, 23], is a cryptographic primitive that can protect confidentiality and integrity at the same time. AEAD takes as input a public nonce IV, public associated data AD, plaintext P, and a key K, outputs the ciphertext C and a tag T while encryption, and while decryption, K, IV, AD, C and T are inputs, if the tag T is valid, returns P, otherwise an error symbol $\perp$.

In 2013, the international cryptologic research community announced a new competition for authenticated encryption called CAESAR, and in August 2016, 15 candidates were elected into the third round, including JAMBU by Wu and Huang [31]. JAMBU (originally AES-JAMBU) is a block-cipher mode of operation, which is a primary method for implementing authenticated encryption. Among other AEAD schemes using modes of operation in CAESAR, JAMBU is designed for lightweight applications. It is not as fast as the parallelizable schemes such as OCB [26] and OTR [17], but it is inverse-free, using only X-OR operations, and has a lower state size in the cost of a shorter nonce and tag length [31]. JAMBU adopts an underlying block-cipher with $2n$-bit block length and k-bit key length, along with n-bit nonce, and outputs n-bit tag. It

[⋆] E-mail: cnpkw@126.com

has only $3n$-bit state, which memory requirement is among the least of CAESAR candidates.

Initial vector (IV), or usually called nonce to show its non-repeatedness, has been important in symmetric key cryptography since the invention of CBC mode. The importance of nonce has been discussed by earlier researchers, especially in the terms of AEAD [8, 24]. Each nonce was supposed to be used only once, but due to various reasons including incorrect implementation, resource limitation, loss of stored nonce data, etc, it is possible that an encryption algorithm returns two ciphertexts with a same nonce, which is often called *nonce misuse*. Most earlier AEAD modes of operation were not designed to support nonce misuse. For example, in GCM mode of operation which is widely adopted as a standard [20], the security would be completely broken if nonce could be reused. But later, especially as the CAESAR competition went on, the community and AEAD designers were divided into two groups: some of them believe nonce should never be reused, and others believe that nonce misuse is inevitable, so an AEAD scheme should at least provide some security when the user repeats a nonce.

The idea of AE with nonce misuse security, has first been introduced with the term deterministic AE [28] or misuse-resistance AE [27]. Researchers also provided some modes of operation which support nonce misuse resistance, such as [15, 29, 12]. However, such security notion requires that there is no information leakage even when nonce is reused, which is sometimes too strong. So a weaker notion of online AE has been studied, often called online nonce misuse resistance AE [11, 1, 13]. An AE scheme is called online, if each output block is only related to its previous input blocks. A perfectly secure online scheme should only leak the common prefix of the message. In the CAESAR competition, there are also some online nonce misuse resistance schemes, for example COLM [2] that has entered the third round.

Although the necessity of nonce misuse security is still under controversy, it is indeed useful for lightweight applications. For protecting confidentiality and integrity in a resource restrained device such as IoT, RFID card, etc, it is not always possible for storing and managing fresh nonce, and sometimes it requires additional synchronous protocol which might be costly. But if the scheme is nonce-misuse resistance, even only to a small degree, then any random number could be used as a nonce, which will simplify the implementation a lot.

In the first version of JAMBU proposal [30], the designers claimed that JAMBU leaks only the common prefix of the message when nonce is reused. However, JAMBU with nonce misuse had later been analyzed by Peyrin et al [22], and they showed that there is an attack with $O(2^{n/2})$ queries on JAMBU with nonce misuse. The designers acknowledged their work. In their latest document [31], the designers gave a proof on the authenticity of JAMBU, but there are still no results on privacy. However, they believe that JAMBU can achieve some security under nonce misuse, although not as they originally claimed. If JAMBU could be proved to have an $n/2$-bit security under nonce misuse case, although not full security, it could still bring great advantage since JAMBU is designed for lightweight usage.

Provable security is an important method in the research of both public key and symmetric key cryptography. In symmetric key cryptography, provable security is usually applied to modes of operation, which security is reduced to the security of the underlying block cipher. The examples are security proofs to OCB [25] and GCM [21]. Although not necessary, a security proof is often considered a great advantage when evaluating a mode of operation. Most of the CAESAR submissions which are modes of operation had given their security proofs. However, the designers did not give their security proof on JAMBU, and this devaluate their security claims compared to other schemes such as CLOC/SILC [14], which shares the same lightweight feature with JAMBU. In this paper, we shall give security proofs for JAMBU under both nonce respecting and nonce misuse cases, so that the security of JAMBU could be further ensured.

Unlike nonce respecting security, there is no common way to define nonce misuse security. In [13], the authors list out various security notions for AEAD schemes in CAESAR competition. One of them (OAE1d) is as follows: "Leaks equality of block-aligned prefixes and the XOR of the block directly following this prefix. E.g., if C, C' arise from 4-block plaintexts $P = A\|B\|C\|D$ and $P' = A\|B\|E\|F$, we always have $C_1 = C'_1$, $C_2 = C'_2$, and $C_3 \oplus C'_3 = C \oplus E$. Ciphertexts C, C' arising from 4-block plaintexts $P = A\|B\|C\|D$ and $P' = E\|F\|G\|H$ will have $C_1 \oplus C'_1 = A \oplus E$." The authors also pointed out that JAMBU, as well as sponge based schemes such as Keyak[6] or Norx[4], satisfied such security notion for nonce misuse.

In this paper, we shall point out that in these schemes, the $i+1$-th ciphertext block is obtained from the XOR of the $i+1$-th plaintext block and a "keystream" block which is generated by the first i plaintext blocks. So that if the first i plaintext blocks are the same for two different inputs, the first $i + 1$ keystream blocks are the same as well, which leads to the insecurity of the first $i + 1$ ciphertext blocks.

It is hence clear that the security of the AEAD scheme can be derived from the security of all keystream blocks, which can be considered as a certain type of pseudorandom number generator (PRNG), we call it online block-wise PRNG. We shall give out its definition and security notion in this paper. An online block-wise PRNG is not an AE scheme, but it can be used for construction of AE schemes, and we prove the relationship between the confidentiality and integrity of AE scheme and the security of the online block-wise PRNG. We further show that the scheme JAMBU can be constructed from a certain type of online blockwise PRNG, which we called JAMBU-like online blockwise PRNG. Then, we use the game-playing proof method [9] to prove the security of JAMBU-like online blockwise PRNG, hence the security of JAMBU is assured. Since the integrity result has been given by the designers, we mainly focus on the confidentiality result. We show that JAMBU indeed can be proven to have an $n - \log n$-bit security under nonce respecting case, which is close to its birthday bound, and for nonce-misuse case, the security is $n/2$-bit under our model.

Related Works. Our construction of online blockwise PRNG partly covers the notion of sponge-based constructions, which security has been widely discussed before. The provably security of sponge functions has been given in [5], and there are also provable security results on sponge-based AE constructions [19, 3]. Most recently, Daemon et al [10] has given the security result for the full-state keyed duplex construction used in Keyak. We claim that, although there are some similarities, ours is an individual work from these sponge-based provable security results.

The paper organized as follows. In section 2, we simply introduce the JAMBU scheme. In section 3, we define online blockwise PRNG and how to construct AEAD schemes from it. In section 4, we prove the security of JAMBU-like online blockwise PRNG, which leads to the security of JAMBU, under both nonce respecting case and nonce misuse case. And finally in section 5 we draw the conclusion.

2 The JAMBU AE Scheme

Notations. We shall use $\|$ as the operator for concatenation of two strings, for example, $0\|1$ is the string 01. ε denotes an empty string. For a string s, we write $s|_{i,j}$ to be the substring of s from the i-th bit to the $j-1$-th bit, for example, $01100|_{2,4} = 10$. If $i = 0$, we also write it as $s|_j$, which is the first j bits of s. We also let $s[i] = s|_{ni,n(i+1)}$ be the i-th block of s, n is the block size of JAMBU. $x \overset{\$}{\leftarrow} X$ means that x is randomly chosen from a set X.

Before we go on to the security proof, we first introduce the structure of JAMBU. As a block cipher mode of operation, JAMBU uses an underlying block-cipher with k-bit key and $2n$-bit block length, takes n-bit nonce IV, arbitrary length of associated data AD and plaintext P as input, generates ciphertext C of the same length as P, and n-bit authentication tag T. JAMBU has $3n$-bit internal states U, V and R, each of n-bit. In their submission, the designers denote the states before a block-cipher call by U, V, R and after a block-cipher call by X, Y, R, we adopt this notation in our discussion. We write E_K as the underlying block cipher.

A JAMBU encryption consists of five steps:

(1) Padding. AD and P are done with a 10^* padding. For associated data, a '1' bit is padded followed by the least number of '0' bits to make the length of padded associated data a multiple of n-bit. Then the same padding method is applied to the plaintext.

(2) Initialization. For the n-bit nonce IV, the state $U_{-1}\|V_{-1}$ is set to $0^n\|IV$, and R_{-1} is set to 0^n. Then, set $X_{-1}\|Y_{-1} \leftarrow E_K(U_{-1}\|V_{-1})$, $U_0\|V_0 \leftarrow X_{-1}\|Y_{-1} \oplus 5$, $R_0 \leftarrow U_0$. Note that 5 is written as a binary string $0^{2n-3}101$, so are other numbers below.

(3) Processing of the associated data. For the padded associated data (note that if there is no associated data, padding made it into at least 1 block), it is divided into h blocks $AD[0], ..., AD[h-1]$, and processed as follows:

For $i = 0$ to $h-1$, update the states:

$X_i\|Y_i \leftarrow E_K(U_i\|V_i)$, $U_{i+1} \leftarrow X_i \oplus AD[i]$, $V_{i+1} \leftarrow Y_i \oplus R_i \oplus 1$, $R_{i+1} \leftarrow R_i \oplus U_{i+1}$, $AD[i]$ is the i-th AD block.

The initialization and processing of the associated data step is shown in figure 1.

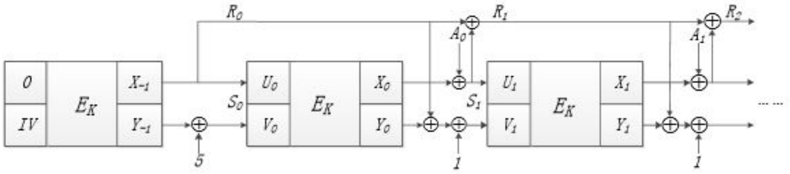

Fig. 1: Initialization and AD processing

(4) Processing of the plaintext. For the padded plaintext, it is divided into p blocks $P[0], ..., P[p-1]$, and processed as follows:

For $i = 0$ to $p-1$ (note that we reset i to 0, and $U_h \| V_h$ in (3) becomes $U_0 \| V_0$), update the states:

$X_i \| Y_i \leftarrow E_K(U_i \| V_i)$, $U_{i+1} \leftarrow X_i \oplus P[i]$, $V_{i+1} \leftarrow Y_i \oplus R_i$, $R_{i+1} \leftarrow R_i \oplus U_{i+1}$, output $C[i] \leftarrow P[i] \oplus V_{i+1}$, $P[i]$ is the i-th plaintext block.

For the last ciphertext block $C[p-1]$, truncate it into the same length as the last plaintext block before padding. For example, if plaintext length is a multiple of n (in this case, a full padding block is added), then the last ciphertext block is simply ignored.

The processing of the plaintext step is shown in figure 2.

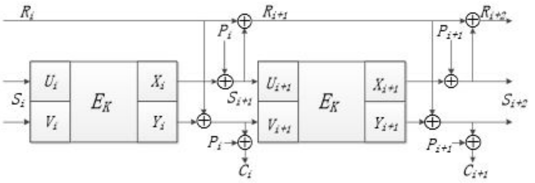

Fig. 2: Plaintext processing

(5) Finalization and tag generation. It process as follows:

$X_p \| Y_p \leftarrow E_K(U_p \| V_p)$, $U_{p+1} \leftarrow X_p$, $V_{p+1} \leftarrow Y_p \oplus R_p \oplus 3$, $R_{p+1} \leftarrow R_p \oplus U_{p+1}$, $X_{p+1} \| Y_{p+1} \leftarrow E_K(U_{p+1} \| V_{p+1})$, output $T \leftarrow X_{p+1} \oplus Y_{p+1} \oplus R_{p+1}$.

The finalization and tag generation step is shown in figure 3.

In a JAMBU decryption, first do the padding, initialization and AD processing step. Then, generate the state V_1, use it to recover $P[0]$, and use $P[0]$ the same way as the plaintext processing step to generate V_2, then recover $P[1]$, etc. After generate a tag T', compare T' with T, if $T' = T$, output the plaintext, otherwise output $\perp$.

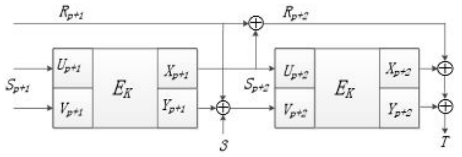

Fig. 3: Finalization and tag generation

3 Online Blockwise PRNG

3.1 Definition and Security Model

Definition 3.1. *Let $\mathcal{I} = \mathcal{N} \cup \mathcal{M} \cup \{eoi\}$, where $\mathcal{N} \cap \mathcal{M} = \emptyset$ and eoi is a special element which $eoi \notin \mathcal{N} \cup \mathcal{M}$. $\mathcal{N}$ is called the nonce base and $\mathcal{M}$ is the set of input blocks.*

Let $len : \mathcal{M} \cup \{eoi\} \mapsto \{0, 1, ..., b\}$, b is any fixed integer, and let $g : \mathcal{N} \times \mathcal{M}^* \mapsto \{0, 1\}^b$ be any function.

Then, an online block-wise PRNG (OBP for short) $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len} : \mathcal{N} \times \mathcal{M}^* \times \{eoi\} \mapsto \{0, 1\}^*$ is defined as:

$$
\begin{aligned}
&\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}(N, M_1, ..., M_m, eoi) \\
&\qquad = g(N)|_{len(M_1)} \| g(N, M_1)|_{len(M_2)} \| ... \| g(N, M_1, ..., M_m)|_{len(eoi)}.
\end{aligned}
$$

We denote $g(N, ..., M_{i-1})|_{len(M_i)}$ by O_i, and $g(N, ..., M_m)|_{len(eoi)}$ by T, so the output can be written as $O_1\|...\|O_m\|T$, and O_i is only related to N and the first $i-1$ input blocks and $len(M_i)$. Sometimes, we can simply write $\mathcal{G}^g$ instead of $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}$ if there is no confusion.

Specifically, if we replace g by a random oracle \$, then $\mathcal{G}^{\$}_{\mathcal{N},\mathcal{M},len}$ is called a random online block-wise oracle.

For the simplicity of further discussion, for any input $(N, M_1, ..., M_m, eoi)$, we also write $M_{m+1} = eoi$ and $O_{m+1} = T$.

The theorem below shows that any output of a random online blockwise oracle can be divided into two parts: the first part is a prefix of some former output, the second par is a uniformly random string.

Theorem 3.1. *Suppose that a random online block-wise oracle $\mathcal{G}^{\$}_{\mathcal{N},\mathcal{M},len}$ has been queried q times, and the j-th query is $Q^j = (N^j, M^j_1, ..., M^j_{m^j}, M^j_{m^j+1})$, $1 \leq j \leq q$. (Note that $M^j_{m^j+1} = eoi$.) We write $Len = \Sigma_{i=1}^{m^j+1} len(M^j_i)$. Then for each query Q^j, either:*

(1) For any $j^* < j$, $N^{j^*} \neq N^j$. In this case, we have $\mathcal{G}^{\$}(Q^j) \xleftarrow{\$} \{0, 1\}^{Len}$; Or

(2) There exists $j' < j$, $1 \leq k \leq m^j$ which $N^j = N^{j'}$, $M^j_i = M^{j'}_i$ for any $i < k$, such that for any $j^* < j$ which $N^{j^*} = N^j$, either there is some $i^* < k$ such

that $M^j_{i^*} \neq M^{j^*}_{i^*}$*; or* $M^j_k \neq M^{j^*}_k$ *and* $len(M^{j^*}_k) \leq len(M^{j'}_k)$*. In this case, let* $pf = \Sigma_{i=1}^{k-1} len(M^j_i) + \min\{len(M^{j'}_k), len(M^j_k)\}$*, then we have* $\mathcal{G}^\$(Q^j) = \mathcal{G}^\$(Q^{j'})|_{pf}\|x$*,* $x \xleftarrow{\$} \{0,1\}^{Len-pf}$*.*

Proof. We use $\mathcal{O}$ to denote the random oracle used in $\mathcal{G}^\$_{\mathcal{N},\mathcal{M},len}$. If for any $j^* < j$, $N^{j^*} \neq N^j$, then $\mathcal{O}(N^j, M^j_1, ..., M^j_i)$ was never queried before, so the output is uniformly random, and $\mathcal{G}^\$(Q^j)$ is also a uniformly random string.

Otherwise, we find k and j' such that $(N^{j'}, M^{j'}_1, ..., M^{j'}_{k-1}) = (N^j, M^j_1, ..., M^j_{k-1})\}$, $len(M^{j'}_k) = \max\{len(M^{j^*}_k)|(N^{j^*}, M^{j^*}_1, ..., M^{j^*}_{k-1}) = (N^j, M^j_1, ..., M^j_{k-1})\}$; and there is no $j^* < j$ where $(N^{j^*}, M^{j^*}_1, ..., M^{j^*}_{k-1}, M^{j^*}_k) = (N^j, M^j_1, ..., M^j_{k-1}, M^j_k)$.

So for $1 \leq i \leq k-1$, $O^j_i = \mathcal{O}(N^j, ..., M^j_{i-1})|_{len(M^j_i)} = \mathcal{O}(N^{j'}, ..., M^{j'}_{i-1})|_{len(M^{j'}_i)} = O^{j'}_i$; for $k+1 \leq i \leq m^j + 1$, $O^j_i = \mathcal{O}(N^j, ..., M^j_{i-1})|_{len(M^j_i)}$ has never been queried before, and is uniformly random. For $i = k$, we can see that although $\mathcal{O}(N^j, ..., M^j_{k-1})$ has been queried before, but only its first $len(M^{j'}_k)$ bits has been output, and its other bits can be viewed as uniformly random. So if $len(M^j_k) \leq len(M^{j'}_k)$, $O^j_k = \mathcal{O}(N^j, ..., M^j_{k-1})|_{len(M^j_k)} = O^{j'}_k|_{len(M^j_k)}$, otherwise, the first $len(M^{j'}_k)$ bits of O^j_k is $O^{j'}_k$, and the rest is uniformly random.

By our discussion above, the first $pf = \Sigma_{i=1}^{k-1} len(M^j_i) + \min\{len(M^{j'}_k), len(M^j_k)\}$ bits of $\mathcal{G}^\$(Q^j)$ is the same as the first pf bits of $\mathcal{G}^\$(Q^{j'})$, and the rest bits are uniformly random. □

Random online block-wise oracle can be used as the ideal security model, and the security of an online block-wise PRNG is defined as the distinguishing advantage between it and random online block-wise oracle. We call it the **obp**-security, which is defined as:

Definition 3.2. *The **obp**-security of an online block-wise PRNG where g is randomly chosen from a set G is defined as:*

$$\mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A}) = |Pr[g \xleftarrow{\$} G : \mathcal{A}^{\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}} \Rightarrow 1] - Pr[\mathcal{A}^{\mathcal{G}^\$_{\mathcal{N},\mathcal{M},len}} \Rightarrow 1]|.$$

Also we write $\mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)} = \max_{\mathcal{A}} \mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A})$*, which is the maximal advantage over all possible adversaries.*

Here, G is the set of all possible functions of g. For example, g is a keyed function which can be written as g_K, and $\mathcal{K}$ is the key space, we can take $G = \{g_K | K \in \mathcal{K}\}$.

Lemma 3.1. *Suppose that adversary* $\mathcal{A}$ *queries* $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}$ *with* $\{Q^j = (N^j, M^j_1, ..., M^j_{m^j}, eoi)|1 \leq j \leq q\}$*. If for each* $1 \leq j \leq q$*,* $0 \leq i \leq m^j$*,* $g \xleftarrow{\$} G : g(N^j, ..., M^j_i)$ *is uniformly random, then* $\mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A})$ *is negligible.*

Proof. If $\mathcal{A}$ queries $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}$ with $\{Q^j = (N^j, M^j_1, ..., M^j_{m^j}, eoi)|1 \leq i \leq q\}$, then it queries g with $\{(N^j, ..., M^j_i)|1 \leq j \leq q, 0 \leq i \leq m^j\}$, which returns the same as \$. By definition of **obp**-security, the advantage for $\mathcal{A}$ is negligible. □

In some AEAD schemes, the unique nonce security can be extended to unique nonce-AD pair security, where the security of the scheme is granted when there are no inputs with same nonce and same associated data. This is because that there is no output when processing associated data. In the terms of online block-wise PRNG, we can also define such kind of security. We use the term *header* for input blocks at the head of the sequence with no output, and formalize it as below:

Definition 3.3. *Let $\mathcal{G}$ be an online bit-wise PRNG. For any query $(N, M_1, ...,$ $M_m, eoi)$ we define its header $H(N, M_1, ..., M_m, eoi) = (N, M_1, ..., M_k)$ where $len(I_i) = 0$ for $i \leq k$, and $len(M_{k+1}) > 0$.*

When consider the unique header security, the **obp**-security degenerates into the classical **prf**-security (non-online).

Definition 3.4. *The unique header security of an online block-wise PRNG where g is randomly chosen from a set G is defined as the classical **prf**-security, which is:*

$$\mathbf{Adv}^{\mathbf{prf}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A}) = |Pr[g \xleftarrow{\$} G : \mathcal{A}^{\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}} \Rightarrow 1] - Pr[\mathcal{A}^{\$} \Rightarrow 1]|.$$

Also we write $\mathbf{Adv}^{\mathbf{prf}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)} = \max_{\mathcal{A}} \mathbf{Adv}^{\mathbf{prf}}_{\mathcal{G}_{\mathcal{I},len}(G)}(\mathcal{A})$, where $\mathcal{A}$ is chosen from the set of all adversaries which never queries $\mathcal{G}$ with two inputs of same header.

Corollary 3.1. *If $\mathcal{A}$ queries with unique header, then $\mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A}) = \mathbf{Adv}^{\mathbf{prf}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A})$.*

Proof. We only need to show that $\mathcal{G}^{\$}_{\mathcal{N},\mathcal{M},len}$ always returns a random string if the adversary never queries it with same header. For any query $Q^j = (N^j, M^j_1, ...,$ $M^j_{m^j}, eoi)$, if there is no previous query $j^* < j$ such that $N^{j^*} = N^j$, then the output is uniformly random.

Otherwise, we find k and j' that satisfies Theorem 3.1. Suppose that $H(N^j, ...,$ $M^j_{m^j}, eoi) = (N^j, ..., M^j_{k'})$. We see that $k' \geq k-1$, otherwise, $(N^j, ..., M^j_{k'}) =$ $(N^{j'}, , ..., M^{j'}_{k'})$ and $len(M^{j'}_{k'+1}) = len(M^{j'}_{k'+1}) > 0$, then $H(Q^{j'}) = (N^j, ..., M^j_k) =$ $H(Q^j)$, and that makes a contradiction. If $k' > k-1$, by Theorem 3.1, $\mathcal{G}^{\$}_{\mathcal{N},\mathcal{M},len}$ $(N^j, ..., M^j_{m^j}, eoi)$ is uniformly random. If $k' = k-1$, then $len(M^{j'}_k) = 0$, otherwise $H(Q^{j'}) = (N^j, ..., M^j_{k-1}) = H(Q^j)$. By Theorem 3.1, we see that $\mathcal{G}^{\$}_{\mathcal{I},len}(Q^j)$ is also uniformly random. □

3.2 AEAD from Online Blockwise PRNG

Now, we build a connection between a nonce-based AEAD scheme and our definition of online blockwise PRNG. First, we define an encoding from an AEAD input to an online blockwise PRNG input.

Definition 3.5. *For an AEAD scheme, let its nonce base be $\mathcal{N}$, and tag length be τ. We construct an online blockwise PRNG $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}$ where $len(eoi) = \tau$. Now we build a connection between the AEAD scheme and the online blockwise PRNG by defining an encoding function $enc : \mathcal{N} \times \{0,1\}^* \times \{0,1\}^* \mapsto \mathcal{N} \times \mathcal{M}^*$ which maps any AEAD input (N, AD, P) into the online blockwise PRNG input $(N, M_1, ..., M_m, eoi)$.*

We define a partition function $par : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}^*$. For each AEAD encryption input (N, AD, P), $N \in \mathcal{N}$ and $AD, P \in \{0,1\}^*$, suppose that $par(|AD|, |P|) = (len_1, ..., len_m)$, which satisfies $\Sigma^m_{i=1} len_i = |P|$. Then, (N, AD, P) can be written as $(N, AD, P_1, ..., P_m)$, where $P = P_1 \| ... \| P_m$ and $|P_i| = len_i$. Similarly, for the decryption input (N, AD, C, T), where $|C| = |P|$, it can be written as $(N, AD, C_1, ..., C_m, T)$, $C = C_1 \| ... \| C_m$ and $|C_i| = len_i$, $|P_i| = |C_i|$ for any $1 \le i \le m$.

Then, an encoding function $enc0 : \mathcal{N} \times \{0,1\}^* \times (\{0,1\}^*)^* \mapsto \mathcal{N} \times \mathcal{M}^*$ is valid, if given any AEAD input $(N, AD, P_1, ..., P_m)$, $enc0$ satisfies:

1) For any AEAD input $(N', AD', P'_1, ..., P'_{m'})$, $enc0(N, AD, P_1, ..., P_m) = enc0(N', AD', P'_1, ..., P'_{m'})$ only if $(N, AD, P_1, ..., P_m) = (N', AD', P'_1, ..., P'_{m'})$;

2) Let $enc0(N, AD, P_1, ..., P_m) = (N', M_1, ..., M_{m'})$, then $N' = N$, $m' = m$, and $len(M_i) = |P_i|$;

3) Let $enc0(N, AD, P_1, ..., P_m) = (N, M_1, ..., M_m)$, then for any $0 \le i \le m$, $enc0(N, AD, P_1, ..., P_i) = (N, M_1, ..., M_i)$.

Given a valid encoding $enc0$, enc is defined as:

$$enc(N, AD, P) = (enc0(N, AD, P_1, ..., P_m), eoi)$$

where $(N, AD, P_1, ..., P_m)$ is generated from (N, AD, P) using par as above.

An example of valid encoding is the encoding used in Keyak [6], which maps any (AD, P) pair into a sequence of b-bit blocks that fit into the inner structure of the scheme. In fact, most AEAD scheme contains such an encoding, although sometimes implicitly.

We can see that while encoding an AEAD input (N, AD, P), P is divided into blocks, and each input block M_i is only determined by AD and the first i input blocks (in most cases, it is only related to the i-th input block P_i itself). This definition links the input plaintext blocks of an AEAD scheme with the input blocks of an online block-wise PRNG.

For any online block-wise PRNG and a valid encoding, we can build an AEAD encryption scheme: for the encryption input (N, AD, P), use par to get $(N, AD, P_1, ..., P_m)$, feed the nonce and every input block in $enc(N, AD, P_1, ..., P_m)$ into the OBP scheme; for each output block O_i which length is $len(M_i) = |P_i|$, XOR it onto the plaintext block P_i to get the ciphertext block C_i; finally feed the symbol *eoi* and take the output as the authentication tag T. The decryption procedure is a bit more tricky. We give the formal definition of an OBP-based AEAD below:

Definition 3.6. *Let $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}$ be an online block-wise PRNG, and enc is a valid encoding. We suppose that $\mathcal{N}$ is also the nonce base of the AEAD scheme, and $len(eoi) = t$ is the tag length. The AEAD scheme $(\mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}, \mathcal{D}^g_{\mathcal{N},\mathcal{M},len,enc})$ is defined by:*

For encryption input (N, AD, P), $\mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}(N, AD, P)$ is generated by:

$$\mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}(N, AD, P) =$$
$$(\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}(enc(N, AD, P))|_{|P|} \oplus P, \mathcal{G}^g_{\mathcal{N},\mathcal{M},len}(enc(N, AD, P))|_{|P|,|P|+t}).$$

For decryption input (N, AD, C, T), the output $\mathcal{D}^g_{\mathcal{N},\mathcal{M},len,enc}(N, AD, C, T)$ is generated by the following procedure:

1) First we use par to get $(N, AD, C_1, ..., C_m, T)$, let $P' \leftarrow \varepsilon$, $i \leftarrow 1$;

2) $P'_i \leftarrow g(enc0(N, AD, P'_1, ..., P'_{i-1}))|_{|C_i|} \oplus C_i$, $P' \leftarrow P' \| P'_i$, $i \leftarrow i + 1$;

3) If $i \leq m$, returns to 2). Let $T' = g(enc0(N, AD, P'_1, ..., P'_m))|_\tau$, if $T' = T$, returns P', otherwise returns $\perp$.

Theorem 3.2. *The AEAD scheme defined above is a correct encryption scheme.*

Proof. For an encryption input (N, AD, P) and its output (C, T), we use *par* and *enc* to get input blocks $(P_1, ..., P_m)$ and $(M_1, ..., M_m)$, correspondingly. By the definition of online block-wise oracle, we have that $C = (g(N)|_{|P_1|} \oplus P_1) \| (g(N, M_1)|_{|P_2|} \oplus P_2) \| ... \| (g(N, M_1, ..., M_{m-1})|_{|P_m|} \oplus P_m)$, and $T = g(N, M_1, ..., M_m)|_t$.

While (N, AD, C, T) is used as decryption input, we can see that $|C_i| = |P_i|$ and $C_i = g(N, M_1, ..., M_{i-1})|_{|P_i|} = g(enc0(N, AD, P_1, ..., P_{i-1}))|_{|P_i|} \oplus P_i$. For $i = 1$, we have that $P'_1 = g(N)|_{|P_1|} \oplus P_1 \oplus g(N)|_{|C_1|} = P_1$. Suppose that for all $i < k$, we have $P'_i = P_i$. Then for $i = k$, we have $P'_k = g(enc0(N, P_1, ..., P_{k-1}))|_{|C_k|} \oplus C_k = P_k$. So $P'_i = P_i$ for all $1 \leq i \leq m$, then $P' = P$. Also, we have $T' = g(enc0(N, AD, P_1, ..., P_m))|_\tau = T$. Then the decryption function outputs $P' = P$, which is the correct decryption result. □

3.3 Security of OBP-based AEAD Schemes

As we mentioned above, for those AEAD schemes which are covered by our definition of OBP-based schemes, if there is a previous query which has a common prefix with the current query, then the output blocks of the common prefix and its next block are insecure. So for the nonce misuse security of OBP-based AEAD schemes, we discard the common prefix blocks and the next block, only discuss the security of other blocks. We define the nonce misuse security by the following definition:

Definition 3.7. *Let $OAE = (\mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}, \mathcal{D}^g_{\mathcal{N},\mathcal{M},len,enc})$ be an OBP-based encryption scheme. Then, the oracle ${\mathcal{E}'}^g_{\mathcal{N},\mathcal{M},len,enc}$ is defined as:*

For any input query (N, AD, P), if the nonce N has never been used before, ${\mathcal{E}'}^g_{\mathcal{N},\mathcal{M},len,enc}(N, AD, P) = \mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}(N, AD, P)$. Otherwise, let $enc(N, AD, P) = (N, M_1, ..., M_m, eoi)$. Among all previous queries, we find the greatest number k such that there exists a previous query (N, AD', P') satisfies $enc(N, AD', P') = (N, M'_1, ..., M'_{m'}, eoi)$ and $M_i = M'_i$ for $i < k$. Let $p = len(M_1) + ... + len(M_k)$. Then if $\mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}(N, AD, P) = (C, T)$, we have ${\mathcal{E}'}^g_{\mathcal{N},\mathcal{M},len,enc}(N, AD, P) = (C|_{p,|P|}, T)$.

The confidentiality of OAE, where g chosen randomly from G is defined as:

$$\mathbf{Adv}^{\mathbf{priv}}_{OAE(G)}(\mathcal{A}) = |Pr[g \xleftarrow{\$} G : \mathcal{A}^{{\mathcal{E}'}^g_{\mathcal{N},\mathcal{M},len,enc}(.,.,.)} \Rightarrow 1] - Pr[\mathcal{A}^{\$(.,.,.)} \Rightarrow 1]|.$$

Here $\$(.,.,.)$ outputs a pair of uniformly random strings of the same length with the output of ${\mathcal{E}'}^g_{\mathcal{N},\mathcal{M},len,enc}(.,.,.)$, and $\mathcal{A}$ is any adversary.

If the adversary is nonce-respecting, which means $\mathcal{A}$ never queries $\mathcal{E}'$ or $\$$ twice with a same N, then $\mathcal{E}'$ is the same as $\mathcal{E}$. Then the confidentiality definition can be also written by:

$$\mathbf{Adv}^{\mathbf{priv}}_{OAE(G)}(\mathcal{A}) = |Pr[g \xleftarrow{\$} G : \mathcal{A}^{\mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}(.,.,.)} \Rightarrow 1] - Pr[\mathcal{A}^{\$(.,.,.)} \Rightarrow 1]|.$$

The two definitions can be considered as nonce-misuse security and nonce-respecting security, respectively. As we already mentioned, the nonce misuse security only consider outputs begin from the $i + 2$-th block if there exists a previous query with the same nonce and a common prefix of i blocks.

Theorem 3.3. *Let $OAE = (\mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}, \mathcal{D}^g_{\mathcal{N},\mathcal{M},len,enc})$ be an OBP-based encryption scheme, where g chosen randomly from G. Then:*

a) $\mathbf{Adv}^{\mathbf{priv}}_{OAE(G)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}$*;*

b) $\mathbf{Adv}^{\mathbf{priv}}_{OAE(G)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{prf}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}$ *for nonce respecting adversary $\mathcal{A}$ which queries $\mathcal{E}^g_{\mathcal{N},\mathcal{M},len,enc}$ with unique nonce.*

Proof. For any adversary $\mathcal{A}$ that attacks the confidentiality of OAE, we construct an adversary $\mathcal{A}'$ that distinguishes between $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}$ and $\mathcal{G}^\$_{\mathcal{N},\mathcal{M},len}$.

$\mathcal{A}'$ acts the same as $\mathcal{A}$ except when $\mathcal{A}$ gets the returned value from oracle calls $\mathcal{E}'$ or $\$$, $\mathcal{A}'$ gets the returned value by:

For $\mathcal{E}'(IV, AD, P) = (C', T')$, suppose that $|C'| = c'$ bits. $\mathcal{A}'$ first call $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}(enc(N, AD, P))$ or $\mathcal{G}^\$_{\mathcal{N},\mathcal{M},len}(enc(N, AD, P))$ and gets an output O of $|P|+t$ bits, then returns $(O|_{|P|-c',|P|} \oplus P|_{|P|-c',|P|}, O|_{|P|,|P|+t})$. We show that the probabilities of $\mathcal{A}'$ and $\mathcal{A}$ returns 1 are the same by proving that the returned value defined above for $\mathcal{A}'$ shares the same distribution with the returned value for $\mathcal{A}$ from $\mathcal{E}'$ or $\$$.

In the real world where $\mathcal{A}$ calls ${\mathcal{E}'}^g_{\mathcal{N},\mathcal{M},len,enc}$ and $\mathcal{A}'$ calls $\mathcal{G}^g_{\mathcal{N},\mathcal{M},len}$, by the definition of $\mathcal{E}$, $\mathcal{E}(IV, AD, P) = (C, T)$ which $(C \oplus P)\|T = \mathcal{G}^g_{\mathcal{N},\mathcal{M},len}(enc(N, AD, P))$. For $\mathcal{A}$, $\mathcal{E}'$ returns the last c' bits of C and T, which is also $O|_{|P|-c',|P|} \oplus P|_{|P|-c',|P|}$ and $O|_{|P|,|P|+t}$, the returned value for $\mathcal{A}'$.

In the ideal world where $\mathcal{A}$ calls \$ and $\mathcal{A}'$ calls $\mathcal{G}^{\$}_{\mathcal{N},\mathcal{M},len}$, by Theorem 2.1, the last $c' + t = len(M_{k+1}) + ... + len(M_m) + len(eoi)$ bits of $\mathcal{G}^{\$}_{\mathcal{N},\mathcal{M},len}$, say, $O_{k+1}, ..., O_m, T$ are uniformly random. For $\mathcal{A}'$, it returns $(O_{k+1} \oplus P_{k+1}) \| ... \| (O_m \oplus P_m) \| T$. By the definition of online block-wise PRNG, M_i must be determined before the output of O_i, so is P_i. Then P_i is independent with O_i, which means that $O_i \oplus P_i$, $i = k+1, ..., m$ and $T = O_{m+1}$ are all uniformly random and independent with previous outputs. Then the return values share the same distribution with $\$(.,.,.)$.

So for each adversary $\mathcal{A}$, we have that $\mathbf{Adv}^{\mathbf{priv}}_{OAE(G)}(\mathcal{A}) = \mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A}') \leq \mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}$.

For the nonce respecting case, from Definition 3.3, $\mathcal{A}'$ always queries $\mathcal{G}$ with unique header. So $\mathbf{Adv}^{\mathbf{prf}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A}')$ is the same as $\mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}(\mathcal{A}')$, and b) can be directly derived from a). □

We shall not discuss the integrity for JAMBU in this paper. However, we still show that our framework of online blockwise PRNG can also be used to analyse the integrity of AEAD schemes.

Definition 3.8. *The integrity of an AE scheme* $OAE = (\mathcal{E}^{g}_{\mathcal{N},\mathcal{M},len,enc}, \mathcal{D}^{g}_{\mathcal{N},\mathcal{M},len,enc})$ *where g is chosen randomly from G is defined as:*

$$\mathbf{Adv}^{\mathbf{auth}}_{OAE(G)}(\mathcal{A}) = Pr[g \xleftarrow{\$} G : \mathcal{A}^{\mathcal{E}^{g}_{\mathcal{N},\mathcal{M},len,enc}(.,.,.),\mathcal{D}^{g}_{\mathcal{N},\mathcal{M},len,enc}(.,.,.,.)} \textit{ forges.}].$$

The restriction on $\mathcal{A}$ is that $\mathcal{A}$ never queries $\mathcal{D}$ with a reply from $\mathcal{E}$.

Definition 3.9. *We say that enc is a prefix encoding, if for any (N, AD, P), $enc(N, AD, P) = (N, M_1, ..., M_m, eoi)$, there is no $(N, AD', P') \neq (N, AD, P)$ such that $enc(N, AD', P') = (N, M_1, ..., M_{m'}, eoi)$, $m' < m$.*

Theorem 3.4. *Suppose that enc is a prefix encoding, τ is the tag length.*

Then, $\mathbf{Adv}^{\mathbf{auth}}_{OAE(G)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)} + q_d/2^t$, *$q_d$ is the number of decryption queries.*

Proof. By Definition 3.6, the tag T is the last t-bit of $\mathcal{G}^{g}_{\mathcal{N},\mathcal{M},len}(enc(N, AD, P))$, and since enc is a prefix encoding, by Theorem 3.1, the last t-bit of $\mathcal{G}^{\$}_{\mathcal{N},\mathcal{M},len}$ $(enc(N, AD, P))$ is a uniformly random string. So if $\mathcal{A}$ can distinguish between T and a uniformly random t-bit string, it can also distinguish between $\mathcal{G}^{g}_{\mathcal{N},\mathcal{M},len}$ and $\mathcal{G}^{\$}_{\mathcal{N},\mathcal{M},len}$, which probability is no more than $\mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)}$ by definition. And if the adversary failed to distinguish between T and a random string, then for each decryption query, the probability of forgery is only the random guessing of $1/2^t$, and for total q_d queries, the probability is no more than $q_d/2^t$. So the total distinguishing probability is no more than $\mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(G)} + q_d/2^t$. □

4 Security of JAMBU-like Online Blockwise PRNG

In this section, we consider JAMBU as an AEAD scheme deduced from an online blockwise PRNG, and prove its security. By our results in the previous section, given the security of this online blockwise PRNG, we can immediately get the security of JAMBU itself.

Compared with the description of the JAMBU mode of operation in section 2, we define a JAMBU-like online blockwise PRNG as follows:

Definition 4.1. *Let $\mathcal{N}, \mathcal{M}$ be the nonce space and message block space, $b = 2n$, $len(eoi) = n$ and for any $M \in \mathcal{M}$, $len(M) \leq n$. Let $\nu : \mathcal{N} \mapsto \{0,1\}^b$ and $\mu : \mathcal{M} \mapsto \{0,1\}^b$ be two injective functions that maps nonce and message block into binary strings. Given any function $f : \{0,1\}^b \mapsto \{0,1\}^b$, the function ρ^f, σ^f, χ^f is recursively defined as:*

(1) $\sigma^f = \nu(N)$, $\rho^f(N) = \nu(N)|_{n,2n}$, $\chi^f(N) = f(\sigma^f(N))$;

(2) $\sigma^f(N, M_1, ..., M_i) = \chi^f(N, M_1, ..., M_{i-1}) \oplus (\rho^f(N, M_1, ..., M_{i-1}) \| 0^n) \oplus \mu(M_i)$, $\rho^f(N, M_1, ..., M_i) = \rho^f(N, M_1, ..., M_{i-1}) \oplus \sigma^f(N, M_1, ..., M_{i-1})|_{n,2n}$, $\chi^f(N, M_1, ..., M_i) = f(\sigma^f(N, M_1, ..., M_i))$ for $1 \leq i \leq m$.

We assume a prefix encoding, so that for any query $(N, M_1, ..., M_m, eoi)$, there is no query $(N, M_1, ..., M_{m'}, eoi)$ such that $m' < m$. Then we can define the keystream function γ_o^f and authentication tag function γ_t^f separately. We let $\gamma_o^f(N, M_1, ..., M_i) = \chi^f(N, M_1, ..., M_i) \oplus (\rho^f(N, M_1, ..., M_i) \| 0^n)$ for $1 \leq i \leq m-1$ and $\gamma_t^f(N, M_1, ..., M_m) = \chi^f(N, M_1, ..., M_i) \oplus (\chi^f(N, M_1, ..., M_i)_{n,2n} \| 0^n) \oplus (\rho^f(N, M_1, ..., M_i) \| 0^n)$, and $\gamma^f = \gamma_o^f \cup \gamma_t^f$.

We say that $\mathcal{G}^{\gamma^f}_{\mathcal{N},\mathcal{M},len}$ is a JAMBU-like online blockwise oracle deduced from f, written as and Γ^f.

Above we construct an online blockwise PRNG which match the definition for JAMBU, where ρ^f is the state R, χ^f is the state $Y \| X$, σ^f is the state $V \| U$. Note that we swap the first n bits and last n bits of each E_K input and output in JAMBU. This does not change the pseudorandomness of E_K (or f in the definition above).

If the function f in Definition 4.1 is a random oracle, then then any interaction between an adversary $\mathcal{A}$ and the JAMBU-like online blockwise PRNG can be simulated by the following interactive game **Game 1** ($\perp(.)$ is a function that always returns $\perp$). We write X instead of $Y \| X$ for simplification, and write V as the intermediate value $Y \oplus R \| X$ (which is also γ_o^f in the definition above).

Since $bad = true$ if and only if there exists i, j such that $bad0_i^j = true$ and $bad1_i^j = false$, it is clear that when $bad = false$, for every i, j such that $bad1_i^j = false$, we have $bad0_i^j = false$. We also write bad_i^j as the intermediate value of bad, which is defined as: $bad_i^j = true$ if and only if there exists $(i', j') < (i, j)$, there is $bad0_{i'}^{j'} = true$ and $bad1_{i'}^{j'} = false$.

Now, we give some lemmas on Game 1, which are useful for our security proof.

Game 1

```
$\pi, \pi' \leftarrow \bot(.)$; $bad \leftarrow false$;
for $j = 1$ to $q$ do
    $S_0^j \leftarrow \nu(N^j)$; $R_0^j \leftarrow S_0^j|_{n,2n}$;
    for $i = 0$ to $m^j$ do
        if $\pi(S_i^j) = \bot$ then $\pi(S_i^j) \xleftarrow{\$} \{0,1\}^b$; $bad0_i^j \leftarrow false$; else $bad0_i^j \leftarrow true$;
        $X_i^j \leftarrow \pi(S_i^j)$;
        if $\pi'(N^j, M_1^j, ..., M_i^j) = \bot$ then $\pi'(N^j, M_1^j, ..., M_i^j) \leftarrow X_i^j$; $bad1_i^j \leftarrow false$; else $bad1_i^j \leftarrow true$;
        if $bad0_i^j = true$ and $bad1_i^j = false$ then $bad \leftarrow true$;
        $V_i^j = X_i^j \oplus (R_i^j \| 0^n)$;
        if $i \neq m^j$ then
            $S_{i+1}^j = V_i^j \oplus \mu(M_{i+1}^j)$;
            output $O_{i+1}^j \leftarrow V_i^j|_{len(M_{i+1}^j)}$;
            $R_{i+1}^j = R_i^j \oplus S_{i+1}^j|_{n,2n}$;
        else output $T^j \leftarrow V_i^j|_n \oplus V_i^j|_{n,2n}$;
        end if
    end for
end for
```

Lemma 4.1. *In Game 1, we have:*

a) $X_i^j = \pi(S_i^j) = \pi'(N^j, M_1^j, ..., M_i^j)$*. Similarly, each value of* S_i^j, R_i^j, V_i^j *are uniquely determined by* $(N^j, M_1^j, ..., M_i^j)$*.*

b) If $bad1_i^j = true$*, then* $bad0_i^j = true$*.*

Proof. In Game 1, if $bad1_i^j = false$, $\pi'(N^j, M_1^j, ..., M_i^j)$ is set to $X_i^j = \pi(S_i^j)$. If $bad1_i^j = true$, then there is a $j' < j$ such that $(N^j, M_1^j, ..., M_i^j) = (N^{j'}, M_1^{j'}, ..., M_i^{j'})$, $\pi'(N^j, M_1^j, ..., M_i^j) = \pi'(N^{j'}, M_1^{j'}, ..., M_i^{j'}) = X_i^{j'}$, and $bad1_i^{j'} = false$, which is the first time that $(N^j, M_1^j, ..., M_i^j)$ occurs in a query. We first prove that for all $k \leq i$, $R_{k-1}^j = R_{k-1}^{j'}$ and $S_k^j = S_k^{j'}$.

We prove it by induction. First, we have $S_0^j = \nu(N^j) = \nu(N^{j'}) = S_0^{j'}$, also $R_0^j = R_0^{j'}$. Now suppose that $R_{k-1}^j = R_{k-1}^{j'}$ and $S_k^j = S_k^{j'}$. Then $R_k^j = R_{k-1}^j \oplus S_k^j|_{n,2n} = R_{k-1}^{j'} \oplus S_k^{j'}|_{n,2n} = R_k^{j'}$; $X_k^j = \pi(S_k^j) = \pi(S_k^{j'}) = X_k^{j'}$; and $S_{k+1}^j = X_k^j \oplus (R_k^j \| 0^n) \oplus \mu(M_{k+1}^j) = X_k^{j'} \oplus (R_k^{j'} \| 0^n) \oplus \mu(M_{k+1}^{j'}) = S_{k+1}^{j'}$ if $k < i$. So $S_i^j = S_i^{j'}$ is already in the domain of π, then $X_i^j = \pi(S_i^j) = \pi(S_i^{j'}) = X_i^{j'} = \pi'(N^j, M_1^j, ..., M_i^j)$, and $bad0_i^j = true$. Also we have $R_i^j = R_i^{j'}$ and $V_i^j = V_i^{j'}$. So for any j and j^* such that $j \neq j^*$ and $(N^j, M_1^j, ..., M_i^j) = (N^{j^*}, M_1^{j^*}, ..., M_i^{j^*})$, there is $S_i^j = S_i^{j'} = S_i^{j^*}$, $X_i^j = X_i^{j'} = X_i^{j^*}$, $R_i^j = R_i^{j'} = R_i^{j^*}$, $V_i^j = V_i^{j'} = V_i^{j^*}$. □

Lemma 4.1 showed that if we use Game 1 to simulate the function $X_i^j = \chi^f(N^j, M_1^j, ..., M_i^j)$ in Definition 4.1, then the function χ^f is well defined, and $\sigma^f, \rho^f, \gamma_o^f$ are all well-defined functions.

Lemma 4.2. *In Game 1, we have:*

a) If $bad0_i^j = false$, $X_i^j, V_i^j, S_{i+1}^j, O_{i+1}^j$(or T^j for $i = m^j$) are uniformly random, and independent with $X_{i'}^{j'}, V_{i'}^{j'}, S_{i'+1}^{j'}, O_{i'+1}^{j'}$(or $T^{j'}$ for $i' = m^{j'}$),$M_{i'+1}^{j'}$ for all $(i',j') < (i,j)$, and $N^{j'}, S_0^{j'}$ for all $j' \leq j$.

b) Suppose that $bad_{i+1}^j = false$. Then X_i^j, V_i^j, O_{i+1}^j(or T^j for $i = m^j$) are uniformly random and independent with $X_{i'}^{j'}, V_{i'}^{j'}, O_{i'+1}^{j'}$(or $T^{j'}$ for $i' = m^{j'}$) for $(i',j') < (i,j)$ such that $(N^j, M_1^j, ..., M_i^j) \neq (N^{j'}, M_1^{j'}, ..., M_{i'}^{j'})$.

Proof. Proof of a): If $bad0_i^j = true$, then X_i^j is assigned as a uniformly random string, which is independent with all previous states and outputs, including $X_{i'}^{j'}, V_{i'}^{j'}, S_{i'+1}^{j'}, O_{i'+1}^{j'}$(or $T^{j'}$), also R_i^j and M_i^j (since M_i^j is chosen by the adversary who knows only all previous outputs, which are independent with X_i^j). Then, using basic results in the probability theory, we have $V_i^j = X_i^j \oplus R_i^j$ and $S_{i+1}^j = X_i^j \oplus R_i^j \oplus \mu(M_i^j)$ are uniformly random and independent with $X_{i'}^{j'}, V_{i'}^{j'}, S_{i'+1}^{j'}, O_{i'+1}^{j'}$, hence $O_{i+1}^j = V_i^j|_{len(M_{i+1}^j)}$ or $T^j = V_i^j|_n \oplus V_i^j|_{n,2n}$ is also uniformly random and independent with $X_{i'}^{j'}, V_{i'}^{j'}, S_{i'+1}^{j'}, O_{i'+1}^{j'}$(or $T^{j'}$).

Also, $M_{i'+1}^{j'}$ and $N^{j'}$ are chosen by the adversary, and only related to the previous outputs. We already shown that $X_i^j, V_i^j, S_{i+1}^j, O_{i+1}^j$(or T^j) are independent with all previous outputs $O_{i^*+1}^{j^*}$(or T^{j^*}), $(i^*, j^*) < (i,j)$. So $X_i^j, V_i^j, S_{i+1}^j, O_{i+1}^j$(or T^j) are independent with $M_{i'+1}^{j'}$ and $N^{j'}$, hence also independent with $S_0^{j'} = \nu(N^{j'})$. □

Proof of b): We find a $j^* \leq j$ such that $(N^j, M_0^j, ..., M_i^j) = (N^{j^*}, M_0^{j^*}, ..., M_i^{j^*})$ and $bad1_i^{j^*} = false$. By our assumption, we have $bad0_i^{j^*} = false$, so using Lemma 4.1, we have $X_i^j = X_i^{j^*}$, $V_i^j = V_i^{j^*}$. Similarly, we can find $j'^* \leq j'$ such that $bad0_{i'}^{j'^*} = false$ and $X_{i'}^{j'} = X_{i'}^{j'^*}$, $V_{i'}^{j'} = V_{i'}^{j'^*}$. As it was proven in a), $X_i^{j^*}, V_i^{j^*}$ are uniformly random and independent with $X_{i'}^{j'^*}, V_{i'}^{j'^*}$ if $j^* \neq j'^*$, which means so are X_i^j, V_i^j and $X_{i'}^{j'}, V_{i'}^{j'}$. We can see that if $(N^j, M_1^j, ..., M_i^j) \neq (N^{j'}, M_1^{j'}, ..., M_{i'}^{j'})$, then $(N^{j^*}, M_1^{j^*}, ..., M_i^{j^*}) \neq (N^{j'^*}, M_1^{j'^*}, ..., M_{i'}^{j'^*})$, so $j^* \neq j'^*$. Then X_i^j, V_i^j, O_{i+1}^j(or T^j) are independent with $X_{i'}^j, V_{i'}^{j'}, O_{i'+1}^{j'}$(or $T^{j'}$) when $(N^j, M_1^j, ..., M_i^j) \neq (N^j, M_1^{j'}, ..., M_{i'}^{j'})$. □

Using Lemma 4.2, we show the independency of $S_i^j, i \geq 1$, $X_i^j, i \geq 0$, $V_i^j, i \geq 0$. The only thing left is the relationship between S_0^j and others.We now estimate the probability that $S_0^j = S_{i'}^{j'}$, $i' > 0$. First we need to define a new function.

Definition 4.2. *Let $r_1, ..., r_y \in \{1, 2, ..., x\}$ be uniformly random and independent variables, and $\lambda_i = |\{r_j | r_j = i\}|$ be the number of variables taken value i. We define $\Lambda(x,y) = \max_{1 \leq i \leq x} \lambda_i$, and $E\Lambda(x,y)$ be the mathematical expectation of $\Lambda(x,y)$.*

Lemma 4.3. *Let $s_1, ..., s_m$ be uniformly random and independent b-bit strings, s is a b-bit string. Suppose that $s_i|_{r,b}$, the last $b-r$ bits of s_i, $1 \leq i \leq m$ are independent with s. Then: $Pr(s \in \{s_1, ..., s_m\}) \leq E\Lambda(2^r, m)/2^{b-r}$.*

Proof. We have $Pr(s = s_i) = Pr(s|_r = s_i|_r)Pr(s|_{r,b} = s_i|_{r,b}) = 2^{-(b-r)}Pr(s|_r = s_i|_r)$. By Definition 4.2, $Pr(s|_r = s_i|_r) \leq \Lambda(2^r, m)/m$. Then $Pr(s \in \{s_1, ..., s_m\}) \leq E\Lambda(2^r, m)/2^{b-r}$. □

Now we are ready to prove the security bounds.

Definition 4.3. *Let $\Gamma^f = \mathcal{G}^{\gamma^f}_{\mathcal{N},\mathcal{M},len}$ be a JAMBU-like online blockwise oracle. Then the **obp**-security of Γ with a random function is defined as: $\mathbf{Adv}^{\mathbf{obp}}_{\Gamma^f}(\mathcal{A}) = \mathbf{Adv}^{\mathbf{obp}}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(\{\gamma^f|f \in Rand(2n)\})}(\mathcal{A})$, where $Rand(2n)$ is the set of functions with $2n$-bit input and $2n$-bit output.*

Theorem 4.1. *a) For the non-unique header case, $\mathbf{Adv}^{\mathbf{obp}}_{\Gamma^f}(\mathcal{A}) \leq q_f^2/2^{2n+1} + qE\Lambda(2^n, q_f)/2^n + qq_f/2^n$;*

b) For the unique header case, $\mathbf{Adv}^{\mathbf{obp}}_{\Gamma^f}(\mathcal{A}) \leq q_f^2/2^{2n+1} + qE\Lambda(2^n, q_f)/2^n$;

Here $q_f = m^1 + ... + m^q + q$ is the total number of input blocks including nonce blocks (excluding eoi).

Proof. By Lemma 4.2, if $bad = false$, we have V_i^j is uniformly random for any i, j. So for $i < m^j$, $\gamma^f(N^j, M_1^j, ..., M_i^j) = V_i^j$ is uniformly random, and for $i = m^j$, $\gamma^f(N^j, M_1^j, ..., M_{m^j}^j) = V_{m^j}^j \oplus (V_{m^j}^j|_{n,2n}\|0^n)$ is also uniformly random (this is because that $P(V) = V \oplus (V|_{n,2n}\|0^n)$ is a permutation for all $2n$-bit strings). By Lemma 3.1, we have that the distinguishing advantage is negligible when $bad = false$, so we only need to calculate the probability of $bad = true$. Now we find an upper bound for it. We can see that $Pr(bad = true) \leq \Sigma_{i,j}Pr(bad0_i^j = true, bad1_i^j = false|bad_i^j = false)$. So we only need to calculate $Pr(bad0_i^j = true, bad1_i^j = false|bad_i^j = false)$ for each i, j.

Since $Pr(bad0_i^j = true, bad1_i^j = false|bad_i^j = false) \leq Pr(bad0_i^j = true|bad1_i^j = false, bad_i^j = false)$, we can use the latter probability for estimation instead. By the description of Game 1, we can see that $Pr(bad0_i^j = true|bad1_i^j = false, bad_i^j = false) \leq \Sigma_{(i',j')<(i,j)}Pr(S_i^j = S_{i'}^{j'}|bad1_i^j = false, bad_i^j = false)$. We divide all S_i^j which $bad1_i^j = false$ and $bad_i^j = false$ into three sets: $\mathcal{S}_1 = \{S_i^j|i > 0 \wedge bad0_{i-1}^j = false\}$; $\mathcal{S}_2 = \{S_i^j|i > 0 \wedge bad0_{i-1}^j = true\}$; $\mathcal{S}_3 = \{S_0^j\}$, and discuss the probability by three cases:

a) $S_i^j \in \mathcal{S}_1$, so $bad0_{i-1}^j = false$. By Lemma 4.2, S_i^j is uniformly random and independent with any $S_{i'}^{j'}$, $(i', j') < (i, j)$. Then $Pr(S_i^j = S_{i'}^{j'}) = 1/2^{2n}$.

b) $S_i^j \in \mathcal{S}_2$, so $bad0_{i-1}^j = true$. Since $bad_i^j = false$, we have $bad1_{i-1}^j = true$. For $(i', j') < (i, j)$, if $(N^j, M_1^j, ..., M_{i-1}^j) = (N^{j'}, M_1^{j'}, ..., M_{i'-1}^{j'})$, then $V_{i-1}^j = V_{i'-1}^{j'}$. But since $bad1_i^j = false$, we have $M_i^j \neq M_{i'}^{j'}$, so $S_i^j \neq S_{i'}^{j'}$.

Otherwise, $(N^j, M_1^j, ..., M_{i-1}^j) \neq (N^{j'}, M_1^{j'}, ..., M_{i'-1}^{j'})$, by Lemma 4.2, V_{i-1}^j is independent with $V_{i'-1}^{j'}$. Because $S_i^j = V_{i-1}^j \oplus \mu(M_i^j)$, also M_i^j is chosen by the adversary and could be written as a function of all previous output. Note that M_i^j must be determined before the output of O_i^j, so M_i^j is related only with outputs from O_1^1 to O_{i-1}^j. We only need to give the independency results between V_{i-1}^j and $O_{i^*}^{j^*}$ (or T^{j^*}) for all $(i^*, j^*) < (i,j)$.

If $(N^j, M_1^j, ..., M_{i-1}^j) \neq (N^{j^*}, M_1^{j^*}, ..., M_{i^*-1}^{j^*})$, by Lemma 4.2, V_{i-1}^j is independent with $O_{i^*}^{j^*}$ (or T^{j^*}). If $(N^j, M_1^j, ..., M_{i-1}^j) = (N^{j^*}, M_1^{j^*}, ..., M_{i^*-1}^{j^*})$, we have $V_{i-1}^j = V_{i^*-1}^{j^*}$, and by our assumption, this output block cannot be the authentication tag. So $O_{i^*}^{j^*}$ is the first $len(M_{i^*}^{j^*})$-bit substring of $V_{i^*-1}^{j^*} = V_{i-1}^j$. If $len(M_{i^*}^{j^*}) = 0$, then $O_{i^*}^{j^*} = \varepsilon$, then V_{i-1}^j is independent with $O_{i^*}^{j^*}$. Otherwise, we still have $len(M_{i^*}^{j^*}) \leq n$. Because V_{i-1}^j is uniformly random, we have that the last n-bit substring of V_{i-1}^j, which is $V_{i-1}^j|_{n,2n}$ is uniformly random and independent with the first $len(M_{i^*}^{j^*})$-bit substring of $V_{i^*-1}^{j^*}$, which is $O_{i^*}^{j^*}$.

If the adversary always queries with unique header, then $len(M_{i^*}^{j^*}) = 0$ for all $(N^j, M_1^j, ..., M_{i-1}^j) = (N^{j^*}, M_1^{j^*}, ..., M_{i^*-1}^{j^*})$. So V_{i-1}^j is independent with all previous outputs (from O_1^1 to O_{i-1}^j), hence V_{i-1}^j is independent with M_i^j. So $S_i^j = V_{i-1}^j \oplus \mu(M_i^j)$ is uniformly random and independent with $S_{i'}^{j'}$, and $Pr(S_i^j = S_{i'}^{j'}) = 1/2^{2n}$.

If the adversary does not query with unique header, we still have $V_{i-1}^j|_{n,2n}$ is independent with all previous outputs, hence independent with M_i^j. Then $S_i^j|_{n,2n} = V_{i-1}^j|_{n,2n} \oplus \mu(M_i^j)|_{n,2n}$ is uniformly random and independent with all previous states. So $Pr(S_i^j|_{n,2n} = S_{i'}^{j'}|_{n,2n}) \leq 1/2^n$, which means $Pr(S_i^j = S_{i'}^{j'}) \leq 1/2^n$ for all $(i', j') < (i, j)$.

c) $S_i^j \in \mathcal{S}_3$, so $i = 0$. Since $bad1_0^j = false$, $N^j \neq N^{j'}$ for all $j' < j$, we have $S_0^j \neq S_0^{j'}$. If the adversary always queries with unique header, then $S_{i'}^{j'}$ is uniformly random for $i' > 0$. Like we discussed in case b), $S_{i'}^{j'}|_{n,2n}$ is independent with all outputs, hence independent with N^j and $S_0^j = \nu(N^j)$. By Lemma 4.3, we have $Pr(S_0^j \in \mathcal{S}_1 \cup \mathcal{S}_2) \leq E\Lambda(2^n, |\mathcal{S}_1 \cup \mathcal{S}_2|)/2^n \leq E\Lambda(2^n, q_f)/2^n$.

If the adversary does not query with unique header, $S_{i'}^{j'}$ is still uniformly random if $S_{i'}^{j'} \in \mathcal{S}_1$, so by Lemma 4.3, we have $Pr(S_0^j \in \mathcal{S}_1) \leq E\Lambda(2^n, |\mathcal{S}_1|)/2^n \leq E\Lambda(2^n, q_f)/2^n$. Also, $S_{i'}^{j'}|_{n,2n}$ is uniformly random if $S_{i'}^{j'} \in \mathcal{S}_2$, as we discussed in case b), also independent with S_0^j. So $Pr(S_{i'}^{j'} = S_0^j) \leq 1/2^n$.

Now we calculate the probability. We have $Pr(bad = true) \leq \Sigma_{i,j} Pr(bad0_i^j = true, bad1_i^j = false | bad_i^j = false) = \Sigma_{i,j} Pr(S_i^j \in \{S_{i'}^{j'} | (i', j') < (i, j)\})$. If we consider the unique header security, then $Pr(S_i^j = S_{i'}^{j'}) \leq 1/2^n$ holds for $i > 0$, and $Pr(bad = true) \leq \Sigma_{i>0,j} Pr(S_i^j \in \{S_{i'}^{j'} | (i', j') < (i, j)\}) + \Sigma_j Pr(S_0^j \in \{S_{i'}^{j'} | (i', j') < (i, j)\}) \leq q_f^2/2^{2n+1} + qE\Lambda(2^n, q_f)/2^n$.

For non-unique header security, we have that $Pr(S_i^j = S_{i'}^{j'}) = 1/2^{2n}$, $(i', j') < (i, j)$ except the following cases: (1) $S_i^j \in \mathcal{S}_2$ or $S_i^j \in \mathcal{S}_3, S_{i'}^{j'} \in \mathcal{S}_2$. This occurs only when $S_i^j \in \mathcal{S}_2$ or $S_{i'}^{j'} \in \mathcal{S}_2$, and the total count is no more than $q_f|\mathcal{S}_2|$. For each query $(N^j, M_1^j, ..., M_{m^j}^j, eoi)$, if $bad1_k^j = true$, there exists $j' < j$ such that $(N^j, M_0^j, ..., M_k^j) = (N^{j'}, M_1^{j'}, ..., M_k^{j'})$, so for all $k' < k$, $bad1_{k'}^j = true$. Then, there is at most one S_i^j for each j such that $bad1_{i-1}^j = true$ and $bad1_i^j = false$, which means for each j there is at most one $S_i^j \in \mathcal{S}_2$. Then $|\mathcal{S}_2| \leq q$. In this case, $Pr(S_i^j = S_{i'}^{j'}) \leq 1/2^n$, and the total probability is no more than $qq_f/2^n$. (2) $Pr(S_0^j \in \mathcal{S}_1) \leq E\Lambda(2^n, q_f)/2^n$ for each $1 \leq j \leq q$, and the total probability is no more than $qE\Lambda(2^n, q_f)/2^n$. Then, $Pr(bad = true) \leq q_f^2/2^{2n+1} + qE\Lambda(2^n, q_f)/2^n + qq_f/2^n$. □

Now we discuss the cases where the random function f in Γ^f is replaced by a block cipher (which is supposed to be a pseudorandom permutation), and define its security as:

Definition 4.4. *Let Γ be a JAMBU-like online blockwise PRNG, $Perm(2n)$ is the set of $2n$-bit permutations. Then the **obp**-security of Γ with a secret random permutation is defined as:*

$$\mathbf{Adv}_{\Gamma^p}^{\mathbf{obp}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{G}_{\mathcal{N},\mathcal{M},len}(\{\gamma^p|p \in Perm(2n)\})}^{\mathbf{obp}}(\mathcal{A}).$$

Theorem 4.2. $\mathbf{Adv}_{\Gamma^p}^{\mathbf{obp}}(\mathcal{A}) \leq \mathbf{Adv}_{\Gamma^f}^{\mathbf{obp}}(\mathcal{A}) + q_f^2/2^{2n+1}$.

Proof. See the PRP-PRF transformation lemma [9]. □

Corollary 4.1. *Suppose that E_K in JAMBU can be considered as a pseudorandom permutation. Let q be the total queries, p be the total number of plaintext blocks, h be the total number of AD blocks. Let $M = p + h + 3q$. Then:*

a) $\mathbf{Adv}_{\text{JAMBU}}^{\mathbf{priv}}(\mathcal{A}) \leq \frac{M^2}{2^{2n}} + \frac{qE\Lambda(2^n,M)}{2^n} + \frac{qM}{2^n}$ *for $\mathcal{A}$ with non-unique nonce-AD pair;*

b) $\mathbf{Adv}_{\text{JAMBU}}^{\mathbf{priv}}(\mathcal{A}) \leq \frac{M^2}{2^{2n}} + \frac{qE\Lambda(2^n,M)}{2^n}$ *for $\mathcal{A}$ with unique nonce-AD pair.*

Proof. For the j-th query (IV^j, AD^j, P^j), we suppose that $AD^j = AD_0^j\|...\|AD_{h^j-1}^j$, $P^j = P_0^j\|...\|P_{p^j-1}^j$, each AD_i^j or P_i^j is an n-bit block. We let $\mathcal{N} \in \{0,1\}^n, \mathcal{M} \in \{0,1\}^n \times \{1,2,3,4\}$, enc be defined as: $enc(IV^j, AD^j, P^j) = (IV^j, (0^n, 1), (AD_0^j, 2), ..., (AD_{h^j-1}^j, 2), (P_0^j, 3), ..., (P_{p^j-1}^j, 3), (0^n, 4))$. It can be simply verified that enc is valid, and is a prefix encoding, since $(0^n, 4)$ only appears as its last element.

We let $\nu(IV^j) = IV^j\|0^n$, $\mu(0^n, 1) = 0^{n-3}\|101\|0^n$, $\mu(AD_i^j, 2) = 0^{n-1}\|1\|AD_i^j$, $\mu(P_i^j, 3) = 0^n\|P_i^j$, $\mu(0^n, 4) = 0^{n-2}\|11\|0^n$, $len(I) = n$ for $I = (P_i^j, 3)$ or $I = eoi$, and $len(I) = 0$ for others. So for each (IV^j, AD^j, P^j), there is a total of $h^j + p^j + 3$ input blocks excluding eoi, then for all encryption queries, the number of total input blocks is $h + p + 3q = M$. Also, we construct E'_K as: $E'_K(X) = E_K(X|_{n,2n}\|X|_n)|_{n,2n}\|E_K(X|_{n,2n}\|X|_n)|_n$, which means that we swap the first n-bit and the last n-bit for both input and output. It is obvious that E'_K is still a pseudorandom permutation.

Now we can see that the definition above fits our definition for JAMBU-like online blockwise PRNG perfectly, and the JAMBU scheme can be viewed as an AEAD scheme based on online blockwise PRNG, the JAMBU-like online blockwise PRNG has unique header if and only if JAMBU has unique nonce-AD pair. So we use Theorem 4.1, Theorem 4.2 and Theorem 3.3 to get the security result of JAMBU. □

The exact value for $E\Lambda(x, y)$ is hard to calculate. But it has been proven in [16, 18], if $y/x = c$ is a constant, then $\lim_{x\to\infty} E\Lambda(x, y) = \lceil \log x/ \log\log x\rceil$. So we suggest that $E\Lambda(2^r, q_f) \approx r/\log r$. Although it is not a strict bound, it can still be used to estimate the security result.

We can see that, for nonce misuse case, JAMBU has confidentiality of $n/2$-bit. Since a distinguisher for JAMBU with complexity $O(2^{n/2})$ has been given [22], our security bound is tight. For nonce respecting case, $\mathbf{Adv}^{\mathbf{priv}}_{\mathrm{JAMBU}}(\mathcal{A}) \leq \frac{M^2}{2^{2n}} + \frac{qE\Lambda(M,2^n)}{2^n} \approx \frac{M^2}{2^{2n}} + \frac{qn/\log n}{2^n}$, which means its confidentiality is at about $n - \log n$-bit.

5 Conclusion

In this paper, we presented a new cryptographic model called online block-wise PRNG, and used it to define a new online nonce misuse security notion for authenticated encryption, which is weaker than the existing online AE security, and use it to prove the security of JAMBU under both nonce respecting and nonce misuse cases. Since the designers did not give the security proof in their submission, we believe that our work is an important complement, especially in the nonce misuse case, where the original security claim of designers has been overthrown by other researchers.

In the nonce misuse case, we show that the security is $n/2$ bits, and since there is an attack with $O(2^{n/2})$ queries, this security bound is tight. However for the nonce respecting case, by using standard provable security method, we can only prove a security up to its birthday bound. That does not necessarily mean that there exists an attack with $O(2^n)$ queries. We believe that JAMBU does have a beyond-birthday-bound security for more than n-bit, and this shall be our future work.

We also hope that, by giving security proofs on JAMBU under both nonce respecting and nonce misuse case, we can help the designers of JAMBU, and bring the scheme further to practical use.

References

1. Andreeva E, Bogdanov A, Luykx A, et al. Parallelizable and authenticated online ciphers. International Conference on the Theory and Application of Cryptology and Information Security. Springer Berlin Heidelberg, 2013: 424-443.
2. Andreeva E, Bogdanov A, Datta N, et al. COLM v1. CAESAR competition proposal, 2016.
3. Andreeva E, Daemen J, Mennink B, et al. Security of keyed sponge constructions using a modular proof approach. International Workshop on Fast Software Encryption. Springer Berlin Heidelberg, 2015: 364-384.

4. Aumasson J P, Jovanovic P, Neves S. NORX: parallel and scalable AEAD. European Symposium on Research in Computer Security. Springer International Publishing, 2014: 19-36.
5. Bertoni G, Daemen J, Peeters M, et al. On the indifferentiability of the sponge construction. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, 2008: 181-197.
6. Bertoni G, Daemen J, Peeters M, et al. CAESAR submission: Keyak v2. CAESAR competion proposal, 2016.
7. Bellare M, Namprempre C. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. International Conference on the Theory and Application of Cryptology and Information Security. Springer Berlin Heidelberg, 2000: 531-545.
8. Bellare, M., Rogaway, P.: Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976. Springer, 2000: 317-330.
9. Bellare M, Rogaway P. Code-Based Game-Playing Proofs and the Security of Triple Encryption. IACR Cryptology ePrint Archive, 2004, 2004: 331.
10. Daemen J, Mennink B, van Assche G. Full-state keyed duplex with built-in multi-user support. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, 2017.
11. Fleischmann E, Forler C, Lucks S. McOE: a family of almost foolproof on-line authenticated encryption schemes. Fast Software Encryption. Springer Berlin Heidelberg, 2012: 196-215.
12. Gueron S, Lindell Y. GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015: 109-119.
13. Hoang V T, Reyhanitabar R, Rogaway P, et al. Online authenticated-encryption and its nonce-reuse misuse-resistance. Annual Cryptology Conference. Springer Berlin Heidelberg, 2015: 493-517.
14. Iwata T, Minematsu K, Guo J, et al. CLOC: Authenticated encryption for short input. International Workshop on Fast Software Encryption. Springer Berlin Heidelberg, 2014: 149-167.
15. Iwata T, Yasuda K. HBS: A single-key mode of operation for deterministic authenticated encryption. Fast Software Encryption. Springer Berlin Heidelberg, 2009: 394-415.
16. Kimber A C. A note on Poisson maxima. Zeitschrift fr Wahrscheinlichkeitstheorie und Verwandte Gebiete, 1983, 63(4): 551-552.
17. Minematsu K. AES-OTR v3. CAESAR competion proposal, 2016.
18. Mitzenmacher M. The power of two choices in randomized load balancing. IEEE Transactions on Parallel and Distributed Systems, 2001, 12(10): 1094-1104.
19. Mennink B, Reyhanitabar R, Vizár D. Security of full-state keyed sponge and duplex: applications to authenticated encryption. International Conference on the Theory and Application of Cryptology and Information Security. Springer Berlin Heidelberg, 2014: 465-489.
20. McGrew D, Viega J. The Galois/counter mode of operation (GCM). Submission to NIST. http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/gcm/gcm-spec.pdf, 2004.
21. McGrew D A, Viega J. The security and performance of the Galois/Counter Mode (GCM) of operation. International Conference on Cryptology in India. Springer Berlin Heidelberg, 2004: 343-355.

22. Peyrin T, Sim S M, Wang L, et al. Cryptanalysis of JAMBU. International Workshop on Fast Software Encryption. Springer Berlin Heidelberg, 2015: 264-281.
23. Rogaway P. Authenticated-encryption with associated-data. Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002: 98-107.
24. Rogaway P. Nonce-based symmetric encryption. International Workshop on Fast Software Encryption. Springer Berlin Heidelberg, 2004: 348-358.
25. Rogaway P. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. International Conference on the Theory and Application of Cryptology and Information Security. Springer Berlin Heidelberg, 2004: 16-31.
26. Rogaway P, Bellare M, Black J. OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Transactions on Information and System Security (TISSEC), 2003, 6(3): 365-403.
27. Rogaway P, Shrimpton T. A provable-security treatment of the key-wrap problem. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, 2006: 373-390.
28. Rogaway P, Shrimpton T. Deterministic Authenticated-Encryption. Advances in Cryptology (EUROCRYPT). 2007, 6.
29. Reyhanitabar R, Vaudenay S, Vizár D. Misuse-resistant variants of the OMD authenticated encryption mode. International Conference on Provable Security. Springer International Publishing, 2014: 55-70.
30. Wu H, Huang T. JAMBU Lightweight Authenticated Encryption Mode and AES-JAMBU. CAESAR competition proposal, 2014.
31. Wu H, Huang T. The JAMBU Lightweight Authentication Encryption Mode (v2.1). CAESAR competition proposal, 2016.