JHAE: An Authenticated Encryption Mode Based on JH

Javad Alizadeh[1], Mohammad Reza Aref[1], Nasour Bagheri[2]

[1] Information Systems and Security Lab. (ISSL), Electrical Eng. Department, Sharif University of Technology, Tehran, Iran, Alizadja@gmail, Aref@sharif.edu

[2] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran, NBagheri@srttu.edu

Abstract. In this paper we present JHAE, an authenticated encryption (AE) mode based on the JH hash mode. JHAE is a dedicated AE mode based on permutation. We prove that this mode, based on ideal permutation, is provably secure.

Keywords: Dedicated Authenticated Encryption, Provable Security, Privacy, Integrity

1 Introduction

Privacy and authentication are two main goals in the information security. In many applications this security parameters must be established simultaneously. An cryptographic scheme that provide both privacy and authentication is called authenticated encryption (AE) scheme. Traditional approach for AE is using of generic compositions. In this approach one uses two algorithms that one provides confidentiality and the other one provides authenticity. However, this approach is not efficient for many applications because it requires two different algorithms with two different keys as well as separate passes over the message [2]. Another approach to design an AE is using a block cipher in a special mode that the block cipher is treated as a black box in the mode [10, 12, 14]. The most problem of these modes is the necessity for implementation of the whole block cipher to process each message block which is time/resource consuming.

Dedicated AE schemes resolve the problems of the generic compositions and block cipher based modes. The designing of a dedicated AE has recently received many attentions in cryptography, mostly driven by the NIST-funded CAESAR competition for AE [7]. Some dedicated AE schemes are ASC-1 [8], ALE [6], AEGIS [16], FIDES [5], CBEAM [13] and APE [1]. A common approach to construct a dedicated AE is to iterate a random permutation or random function in a special mode of operation. Therefore there are two main stages in designing a new dedicated AE:

1. Designing a new dedicated mode (based on a random permutation or a random function)
2. Designing a new random permutation or a random function to be used in the mode.

An general approach is to design a dedicated AE mode from a hash function mode. For example the modes of FIDES, CBEAM and APE are obtained from sponge mode [4]. Another examples are FWPAE and FPAE modes that are obtained from FWP and FP hash function modes respectively [9]. An important challenge in developing an AE mode from another mode is to prove its security to ensure that the transient to another application does not make any structural flows.

In this paper we propose a new dedicated AE mode, JHAE. JHAE is a permutation-based AE mode based on JH hash function mode [15]. It is an on-line and single-pass dedicated AE mode that supports optional associated data (AD). JHAE's security relies on using nonces. We prove that the mode achieves privacy (indistinguishability under chosen plaintext attack or IND-CPA) and integrity (integrity of ciphertext or INT-CTXT) up to $O(2^{n/2})$ queries, where the length of the used permutation is $2n$. In addition, we show that the integrity bound of JHAE is reduced to the indifferentiability of JH hash mode which is at least $O(2^{n/2})$. In Table 1, a comparison between JHAE and some other known dedicated AE modes is given.

Table 1. Comparison between JHAE and known dedicated AE modes

Dedicated AE	Provable Security	AD	On-Line	Nonce Misuse Resistance	Inverse-Freeness of π (or f)	Refference
ASC-1	Yes	No	No	No	Yes	[8]
ALE	No	Yes	Yes	No	Yes	[6]
AEGIS	No	Yes	Yes	No	Yes	[16]
FIDES	No	Yes	Yes	No	Yes	[5]
CBEAM	No	Yes	Yes	No	Yes	[13]
APE	Yes	Yes	Enc only	Yes	No	[1]
JHAE	Yes	Yes	Yes	No	Yes	This paper

The paper is structured as follows: section 2 gives a specification of JHAE encryption-authentication and decryption-verification. The security of JHAE is analysed in section 3. In this section we prove privacy of JHAE in the ideal permutation model, using game playing framework [3] and integrity of it by reducing to the security of JH hash mode. Finally we conclude in section 4.

2 JHAE Authenticated Encryption Mode

In this section we describe JHAE mode, depicted in Fig 2. JHAE is developed from JH hash function mode (Fig 1) [15] and iterates a fixed permutation $\pi : \{0,1\}^{2n} \to \{0,1\}^{2n}$. It is a nonce based, single-pass, and an on-line dedicated AE mode that supports AD.

2.1 Encryption and Authentication

JHAE accepts a n-bit key K, a n-bit nonce N, a message M, an optional AD ,A, and produces ciphertext C and authentication tag T. The pseudo code of JHAE's encryption-authentication is depicted in Table 2. We assume that the input message after padding, is a multiple of the block size (n). The padding approach is very simple, includes appending a single bit '1' followed by a sequence of '0' such that the padded message is a multiple of n. If there is AD in the procedure, then it is also padded to be multiple of n and processed in a way similar to the message block with an exception that ciphertext blocks, c_i, are not produced for AD blocks.

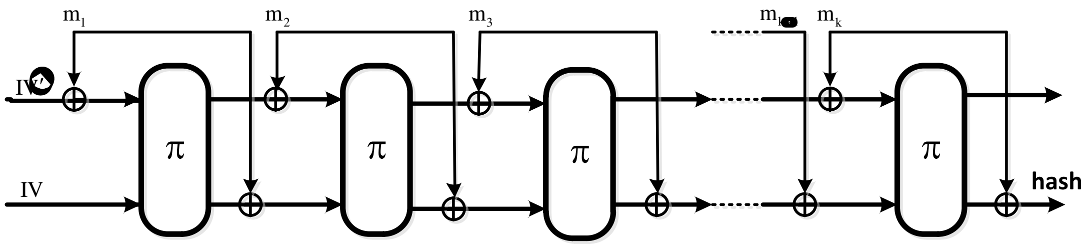

Fig. 1. JH hash mode [11]

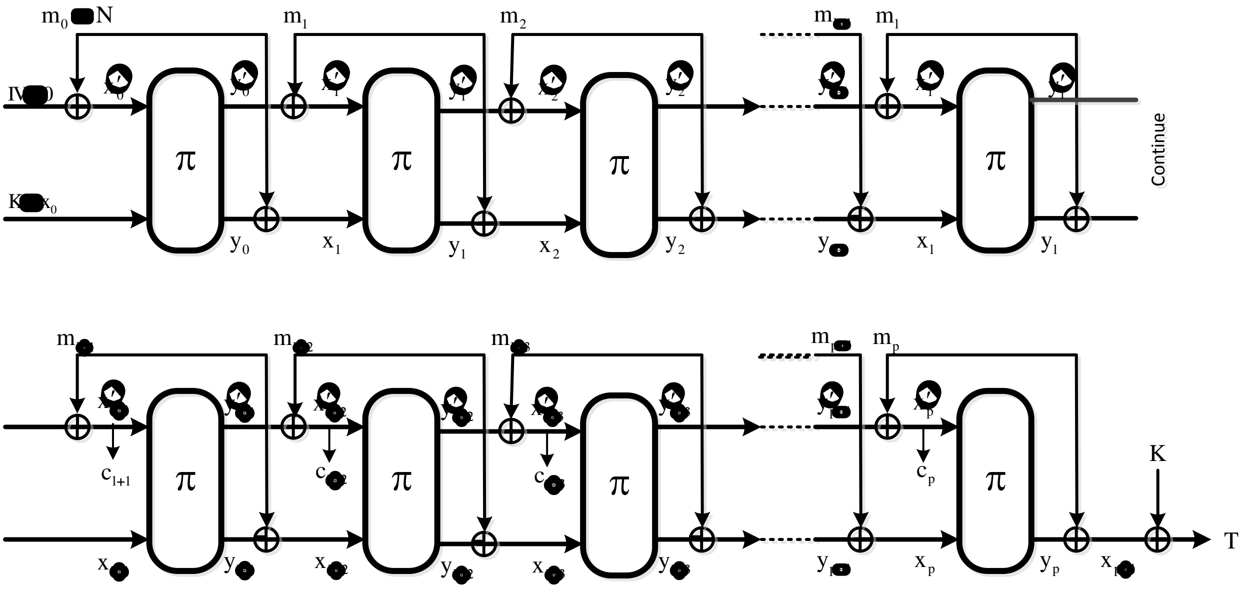

Fig. 2. JHAE mode of operation (encryption and authentication), where $pad(A) = m_1 \| m_2 \| \ldots \| m_l$ and $pad(M) = m_{l+1} \parallel m_{l+2} \parallel ... \parallel m_p$

Table 2. Encryption and authentication pseudo code of JHAE

```
Algorithm1. $JHAE - E^{\pi}(K, N, M, A)$
Input: Key $K$ of $n$ bits, Nonce $N$ of $n$ bits, Associated data $A$ where $pad(A) = m_1 \| m_2 \| \ldots \| m_l$
and Message $M$ where $pad(M) = m_{l+1} \parallel m_{l+2} \parallel ... \parallel m_p$
Output: Ciphertext $C$, Tag $T$
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 0$ to $p - 1$ do:
    $y'_i \parallel y_i = \pi(x'_i \parallel x_i)$;
    $x'_{i+1} = y'_i \oplus m_{i+1}$;
    $x_{i+1} = y_i \oplus m_i$
end for
$y'_p \parallel y_p = \pi(x'_p \parallel x_p)$;
$x_{p+1} = y_p \oplus m_p$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
$T = x_{p+1} \oplus K$
Return $(C, T)$
```

2.2 Decryption and Verification

JHAE decryption-verification procedure, depicted in Table 3, accepts a n-bit key K, a n-bit nonce N, a ciphertext C, a tag T, an optional AD, A, and decrypts the ciphertext to get message M and tag T'. If $T' = T$ then it outputs M else it outputs $\perp$.

3 Security Proofs

In this section we prove the security of JHAE. First we use game playing framework proposed by Bellare and Rogaway [3] and obtain an upper bound for the advantage of an adversary that can distinguish the JHAE from a random oracle (IND-CPA) in the ideal permutation model. Then we prove that JHAE provides integrity (INT-CTXT) until JH hash mode is indifferentiable from a random oracle or tag can not guessed. We follow our proofs in two subsections: privacy and integrity.

3.1 Privacy

In this section, we provide privacy's security bound for the JHAE based on ideal permutation π.

Theorem 1. *JHAE based on an ideal permutation $\pi : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$, is (t_A, σ, ϵ)-indistinguishable from an ideal AE based on a random function RO and ideal permutation π' with the same domain and range, for any t_A then $\epsilon \leq \dfrac{\sigma(\sigma-1)}{2^{2n-1}} + \dfrac{\sigma^2}{2^{2n}} + \dfrac{\sigma^2}{2^n}$, where σ is the total number of blocks in queries to $JHAE - E$, π, and π^{-1}, by $\mathcal{A}$.*

Table 3. Decryption and verification pseudo code of JHAE

```
Algorithm2. $JHAE - D^{\pi}(K, N, C, T, A)$
Input: Key $K$ of $n$ bits, Nonce $N$ of $n$ bits, Associated Data $A$ where $pad(A) = m_1 \| m_2 \| \ldots \| m_l$
, ciphertext $C = c_1 \parallel c_2 \parallel ... \parallel c_p$ and Tag $T$
Output: Message $M$ or $\perp$
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
$x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_{l+p} = c_1 \parallel c_2 \parallel ... \parallel c_p$
for $i = 0$ to $l - 1$ do:
    $y'_i \parallel y_i = \pi(x'_i \parallel x_i)$;
    $x'_{i+1} = y'_i \oplus m_{i+1}$;
    $x_{i+1} = y_i \oplus m_i$
end for
for $i = l$ to $p - 1$ do:
    $y'_i \parallel y_i = \pi(x'_i \parallel x_i)$;
    $m_{i+1} = y'_i \oplus x'_{i+1}$;
    $x_{i+1} = y_i \oplus m_i$
end for
$y'_p \parallel y_p = \pi(x'_p \parallel x_p)$;
$x_{p+1} = y_p \oplus m_p$
$M = m_{l+1} \parallel m_{l+2} \parallel ... \parallel m_p$
$T' = x_{p+1} \oplus K$
if $T' = T$
  Return $M$
else
  Return $\perp$
```

Proof. To the proof the above theorem we use game playing framework based on ten games G_0 to G_9 where G_0 represents JHAE based on ideal permutation π , $JHAE-\pi,\pi^{-1}$, and G_9 represents a random oracle, RO. To determine the adversary's advantage on distinguishing JHAE from an ideal AE scheme, we calculate the adversary's advantage moving from a game to the next game.

Game G_0. This game shows the communication of $\mathcal{A}$ with $JHAE-\pi,\pi^{-1}$ (see Table 4). In this game the permutations π and π^{-1} are exactly the permutations that are used in the real JHAE mode. Hence:

$$Pr[\mathcal{A}^{G_0}=1]=Pr[\mathcal{A}^{JHAE-E}=1]$$

Game G_1. This game is identical to G_0 with an exception that the ideal permutation (π,π^{-1}) is chosen in a "lazy" manner, oracles O_2 and O_3 respectively (see Table 5). These oracles perfectly simulate two ideal permutations and since we assumed that π and π^{-1} in G_0 are ideal permutations then G_0 and G_1 are identical. Therefore we have:

$$Pr[\mathcal{A}^{G_1}=1]=Pr[\mathcal{A}^{G_0}=1].$$

Game G_2. We do a PRP-PRF switch [3] in G_1 and generate G_2 (see Table 6). This means that the ideal permutations O_2 and O_3 in G_1 are replaced with two random functions in G_2. Therefore only difference between G_2 and G_1 is oracles O_2 and O_3 (that in G_1 simulate two ideal permutations but in G_2 simulate two random functions). Unlike the ideal permutation it is possible to find a collision in a random function. Since in G_1 we do not have any collision but in G_2 we may have a collision in O_2 or O_3 the adversary can differentiate G_2 from G_1. Hence, we define a collision in G_2 as a bad event and denote it by bad_0. G_2 and G_1 are identical until bad_0 occurs. Suppose that the adversary can do at most σ_2 and σ_3 query to O_2 and O_3 respectively and let $\sigma'=\sigma_2+\sigma_3$. Then:

$$Pr[\mathcal{A}^{G_2}=1]-Pr[\mathcal{A}^{G_1}=1]=Pr[bad_0\leftarrow true]=Pr[Collision\ in\ O_2\ or\ O_3\ in\ G2]$$

$$\leq\frac{\sigma_2(\sigma_2-1)}{2^{2n+1}}+\frac{\sigma_3(\sigma_3-1)}{2^{2n+1}}\leq\frac{\sigma'(\sigma'-1)}{2^{2n+1}}\leq\frac{\sigma(\sigma-1)}{2^{2n+1}}.$$

Game G_3. In G_3, oracle O_1 does not pass any query to the oracle O_2 but it exactly simulates behavior of oracle O_2(see G_3 Table 7). Thus G_3 and G_2 are identical from adversary's view:

$$Pr[\mathcal{A}^{G_3}=1]=Pr[\mathcal{A}^{G_2}=1].$$

Game G_4. In G_4 (see Table 8) we aim to push the behavior of O_1 one step towards random oracle. Hence, we separate queries that are included to O_2 by O_1 and those that are directly query by the adversary to O_2 or O_3. In this game, if an intermediate query generated by O_1, that is expected to be queried to O_2, has a record in the part of O_2 not included by O_1, it considered as a bad event and denoted by bad_1. However, the distribution of responses of queries to O_2 and O_3 remains identical as G_3. Hence, we can state that G_3 and G_4 are identical until bad_1 occurs in G_4. Assuming that the adversary can do at most σ_1 query to O_1 and σ' query to O_2 or O_3, the adversary's advantage from G_3 to G_4 is bounded as follows:

$$Pr[\mathcal{A}^{G_4} = 1] - Pr[\mathcal{A}^{G_3} = 1] = Pr[bad_1 \leftarrow true] \leq \frac{\sigma'(\sigma_1)}{2^{2n}} \leq \frac{\sigma^2}{2^{2n}}.$$

Game G_5. In G_4 (see Table 9) the responses of O_2 or O_3 are compatible with responses of O_1. In G_5 we aim to push the behavior of O_2 and O_3 one step towards ideal permutations that are independent of RO. For this reason, we generate two auxiliary tables to keep the input and the output of intermediate tentative queries to O_2 generated by O_1 that are denoted by W and Y respectively. Since we aim to do not return any record that has been included to O_2 by O_1 when adversary directly queries to O_2 or O_3 ,in this game, if a query to O_2 or O_3 has a record in W and Y respectively, we considered as a bad event and denote it by bad_2. More precisely, on query to O_1, when it generates local tentative fresh query w_i to O_2 and generate y_i as response, then w_i is stored in W and y_i is stored in Y. However, the distribution of responses to queries to O_1 remains identical as G_4. Hence, we can state that G_4 and G_5 are identical until bad_2 occurs in G_4. To bound the probability of bad_2, suppose that w_j is j-th block that is queried to O_1 and y_j is the response of O_1 to the query where $1 \leq j \leq \sigma_1$ and suppose that v_i is i-th query to O_2 where $1 \leq i \leq \sigma_2$ and z_k is k-th query to O_3 where $1 \leq k \leq \sigma_3$. Then:

$$Pr[bad_2 \leftarrow true] = \sum_{i=1}^{\sigma_2}\sum_{j=1}^{\sigma_1} Pr[v_i = w_j] + \sum_{k=1}^{\sigma_3}\sum_{j=1}^{\sigma_1} Pr[z_k = y_j] \leqslant \frac{\sigma_2\sigma_1}{2^n} + \frac{\sigma_3\sigma_1}{2^n}.$$

It must be noted that in the above calculations we considered the fact that given the response of a query to O_1, the adversary can determine half of the bits of each $w_j \in W$ and $y_i \in Y$. Hence, the adversary's advantage from G_4 to G_5 is bounded as follows:

$$Pr[\mathcal{A}^{G_5} = 1] - Pr[\mathcal{A}^{G_4} = 1] \leq \frac{\sigma_1 \times (\sigma_2 + \sigma_3)}{2^n} \leqslant \frac{\sigma^2}{2^n}.$$

Game G_6. G_6 (see Table 10) is identical as G_5 with an exception that O_1 does not keeps the history of intermediate queries. Hence, in the adversary's view the distribution of the returned values in G_5 and G_6 are identical as far as there is not a intermediate collision in G_5. However, the distribution of responses to queries to O_2 and O_3 remains identical as G_5. Hence, the adversary's advantage from G_5 to G_6 is bounded as follows:

$$Pr[\mathcal{A}^{G_6} = 1] - Pr[\mathcal{A}^{G_5} = 1] \leq \frac{\sigma_1 \times (\sigma_1 - 1)}{2^{2n}} \leq \frac{\sigma \times (\sigma - 1)}{2^{2n}}.$$

Game G_7. In Game G_7 (see Table 11), the blocks of ciphertext and tag value are generated randomly. However, it has now impact of the distribution of the returned values to the adversary. Hence, the distribution of the returned values in G_6 and G_7 are identical:

$$Pr[\mathcal{A}^{G_7} = 1] = Pr[\mathcal{A}^{G_6} = 1].$$

Game G_8. In Game G_8 (see Table 12) we do a PRF-PRP switch [3]. This means that the ideal random functions O_2 and O_3 in G_7 are replaced with a random permutation and its inverse in G_8. Therefore, the only difference between G_7 and G_8 is oracles O_2 and O_3. Thus G_5 and G_4 are identical until O_2 or O_3 has a collision in G_7. Hence, the adversary's advantage from G_7 to G_8 is bounded as follows:

$$Pr[\mathcal{A}^{G_8} = 1] - Pr[\mathcal{A}^{G_7} = 1] = Pr[Collision\ in\ O_2\ or\ O_3\ in\ G_4]$$

$$\leq \frac{\sigma_2(\sigma_2 - 1)}{2^{2n+1}} + \frac{\sigma_3(\sigma_3 - 1)}{2^{2n+1}} \leq \frac{\sigma'(\sigma' - 1)}{2^{2n+1}} \leq \frac{\sigma(\sigma - 1)}{2^{2n+1}}$$

.

Game G_9. In G_8 for each message/AD block a random value is selected and similarly a random value is selected as the tag value. Next these random values are concatenated and returned to the adversary. However, in G_9 (see Table 13) on query to O_1, a random string of the length the desired cipher and tag is selected and returned to the adversary. However, this modification from G_8 to G_9 has no impact on the distribution of the returned values to the adversary. Hence:

$$Pr[\mathcal{A}^{G_9} = 1] = Pr[\mathcal{A}^{G_8} = 1].$$

On the other hand G_8 perfectly simulates RO, π, π^{-1}. Then we have:

$$Pr[\mathcal{A}^{RO,\pi,\pi^{-1}} = 1] = Pr[\mathcal{A}^{G_9} = 1].$$

Finally using of fundamental lemma of game playing [3], we can state:

$$\begin{aligned}
Adv_{JHAE}^{Privacy}(\mathcal{A}) &= Pr[\mathcal{A}^{JHAE-E,\pi,\pi^{-1}} = 1] - Pr[\mathcal{A}^{RO,\pi,\pi^{-1}} = 1] \\
&= Pr[\mathcal{A}^{G_0} = 1] - Pr[\mathcal{A}^{G_9} = 1] \\
&= (Pr[\mathcal{A}^{G_0} = 1] - Pr[\mathcal{A}^{G_1} = 1]) \\
&+(Pr[\mathcal{A}^{G_1} = 1] - Pr[\mathcal{A}^{G_2} = 1]) \\
&+(Pr[\mathcal{A}^{G_2} = 1] - Pr[\mathcal{A}^{G_3} = 1]) \\
&+(Pr[\mathcal{A}^{G_3} = 1] - Pr[\mathcal{A}^{G_4} = 1]) \\
&+(Pr[\mathcal{A}^{G_4} = 1] - Pr[\mathcal{A}^{G_5} = 1]) \\
&+(Pr[\mathcal{A}^{G_5} = 1] - Pr[\mathcal{A}^{G_6} = 1]) \\
&+(Pr[\mathcal{A}^{G_6} = 1] - Pr[\mathcal{A}^{G_7} = 1]) \\
&+(Pr[\mathcal{A}^{G_7} = 1] - Pr[\mathcal{A}^{G_8} = 1]) \\
&+(Pr[\mathcal{A}^{G_8} = 1] - Pr[\mathcal{A}^{G_9} = 1]) \\
\leq 0 + \frac{\sigma(\sigma-1)}{2^{2n+1}} &+ 0 + \frac{\sigma^2}{2^{2n}} + \frac{\sigma^2}{2^n} + \frac{\sigma(\sigma-1)}{2^{2n}} + 0 + \frac{\sigma(\sigma-1)}{2^{2n+1}} + 0 \\
&\leq \frac{\sigma(\sigma-1)}{2^{2n-1}} + \frac{\sigma^2}{2^{2n}} + \frac{\sigma^2}{2^n}.
\end{aligned}$$

3.2 Integrity

In this section, we prove the integrity of ciphertext (INT-CTXT) of JHAE. The INT-CTXT security bound of a permutation based AE scheme is defined as the maximum advantage of any adversary to produce a valid triple $(N, A\|C, T)$ (e.g. a forgery for the AE scheme) without direct query it to the scheme. To forge an AE scheme, the adversary can query to $AE - E$ (the encryption and authentication), $AE - D$ (the decryption and verification), π or π^{-1}. Thus, we can consider two phases for any forgery attempt as follows:

1. **Data gathering:** the adversary gathers some valid triples such as $S = (N_i, (A\|C)_i, T_i); 1 \leq i \leq q$ by at most q queries to $AE - E$, π or π^{-1}.
2. **Execution:** the adversary produces a new triple $(N, A\|C, T)$ such that $(N, A\|C, T) \notin S$ is accepted by $AE - D$ as a valid triple.

In this section, we show that the advantage of any adversary that makes a reasonable number of queries to $JHAE - E$, π, and π^{-1} is negligible in forgery attack against $JHAE$.

Theorem 2. *For any adversary $\mathcal{A}$ that makes in total σ block queries to $JHAE - E$, π, or π^{-1}, JHAE based on an ideal permutation $\pi : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$, is (t_A, σ, ϵ)-unforgeable, for any t_A then $\epsilon \leq \frac{3\sigma^2}{2^n} + \frac{3q}{2^n}$.*

Proof. Suppose that $\mathcal{A}$ is an adversary that tries to forge JHAE. $\mathcal{A}$ should at the first query to JHAE, q times, and produce a list $S = \{(N_i, (A\|C)_i, T_i); 1 \leq i \leq q\}$. Next, $\mathcal{A}$ produces a new $(N, A\|C, T) \notin S$ such that $JHAE - D(N, A\|C, T) \neq \perp$ as its forged triple. All of the possible cases for the new valid $(N, A\|C, T)$ are as follows (case 001 to case 111).

1. **Case 001.** Adversary generates a valid $(N, A\|C, T) \notin S$ such that $\exists (N_i, (A\|C)_i, T_i) \in S : N = N_i, A\|C = (A\|C)_i, T \neq T_i$, for $0 \leq i \leq q$.
2. **Case 010.** Adversary generates a valid $(N, A\|C, T) \notin S$ such that $\exists (N_i, (A\|C)_i, T_i) \in S : N = N_i, A\|C \neq (A\|C)_i, T = T_i$, for $0 \leq i \leq q$.
3. **Case 011.** Adversary generates a valid $(N, A\|C, T) \notin S$ such that $\forall (N_i, (A\|C)_i, T_i) \in S : A\|C \neq (A\|C)_i, T \neq T_i$, for $0 \leq i \leq q$ and $\exists (N_i, (A\|C)_i, T_i) \in S : N = N_i, A\|C \neq (A\|C)_i, T \neq T_i$.
4. **Case 100.** Adversary generates a valid $(N, A\|C, T) \notin S$ such that $\exists (N_i, (A\|C)_i, T_i) \in S : N \neq N_i, A\|C = (A\|C)_i, T = T_i$, for $0 \leq i \leq q$.
5. **Case 101.** Adversary generates a valid $(N, A\|C, T) \notin S$ such that $\exists (N_i, (A\|C)_i, T_i) \in S : N \neq N_i, A\|C = (A\|C)_i, T \neq T_i$, for $0 \leq i \leq q$.
6. **Case 110.** Adversary generates a valid $(N, A\|C, T) \notin S$ such that $\exists (N_i, (A\|C)_i, T_i) \in S : N \neq N_i, A\|C \neq (A\|C)_i, T = T_i$, for $0 \leq i \leq q$.
7. **Case 111.** Adversary generates a valid $(N, A\|C, T) \notin S$ such that $\forall (N_i, (A\|C)_i, T_i) \in S : N \neq N_i, A\|C \neq (A\|C)_i, T \neq T_i$, for $0 \leq i \leq q$.

Hence, we can upper bound the adversary's advantage to forge JHAE as follows:

$$\begin{aligned} Pr[\mathcal{A}^{INT}_{JHAE} = 1] = Pr[Case\ 001] + Pr[Case\ 010] + Pr[Case\ 011] \\ +Pr[Case\ 100] + Pr[Case\ 101] + Pr[Case\ 110] + Pr[Case\ 111]. \end{aligned} \quad (1)$$

To determine an upper bound for this advantage, we categorize the mentioned cases as three distinct sets as follows and determine the adversary's advantage to produce a successful forgery for each set.

Set 1: Set 1 includes any case that could not be used to forge JHAE successfully. More precisely, any triple that matches to the case 001 can not be used to forge JHAE. The reason comes from the fact that for JHAE, for a valid triple, if $A\|C = (A\|C)_i$ and $N = N_i$ then $T = T_i$. Therefore:

$$Pr[Case\ 001] = 0.$$

Set 2: Set 2 includes any case that can be directly used to differentiate JH hash mode from a random oracle. To determine these cases, we consider JH hash mode in Fig 1. Since $T = T_i$ (for $1 \leq i \leq q$) implies $(x_{p+1})_i = (x_{p+1})$ and $(x_{p+1})_i$ and (x_{p+1}) are hash outputs in JH hash mode, then the cases 010, 100, and 110 in the forgery attempt of JHAE lead to collisions in JH hash mode. In other words if the cases 010, 100, and 110 occur in the forgery attempt of JHAE, one can find a collision in JH hash mode and therefore differentiate the mode from a random oracle. Since the bound of the indifferentiability of JH has been proved to be $\frac{\sigma^2}{2^n}$ [11] then:

$$Pr[Case\ 010] = Pr[Case\ 100] = Pr[Case\ 110] \leq \frac{\sigma^2}{2^n}.$$

Set 3: This set include cases that forces the adversary to guess the tag. More precisely, in the cases 011, 101 and 111 the adversary finds a new valid $(N, A\|C, T)$ such that $\forall (N_i, (A\|C)_i, T_i) \in S : N \neq N_i$ or $A\|C \neq (A\|C)_i$. On the other hand, given such a pair of N and $A\|C$, the distribution of the valid tag would be uniformly distributed over $\{0,1\}^n$. Hence, on each attempt, the adversary's advantage to generate a valid tag would be 2^{-n}. So:

$$Pr[Case\ 101] = Pr[Case\ 011] = Pr[Case\ 111] \leq \frac{q}{2^n}$$

Finally using Equation 1 we have:

$$Pr[\mathcal{A}^{INT}_{JHAE} = 1] \leq \frac{3\sigma^2}{2^n} + \frac{3q}{2^n}$$

4 Conclusion

In this paper we introduce JHAE, a new dedicated permutation-based AE mode. In the ideal permutation model, we proved that JHAE provides IND-CPA and INT-CTXT up to $q = O(2^{n/2})$. For a future work one can design a new permutation and implement it by JHAE mode.

References

1. E. Andreeva, B. Bilgin, A. Bogdanov, A. Luykx, B. Mennink, N. Mouha, and K. Yasuda. APE: Authenticated Permutation-Based Encryption for Lightweight Cryptography. 2014.
2. M. Bellare and C. Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. *J. Cryptology*, 21(4):469–491, 2008.
3. M. Bellare and P. Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.
4. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Sponge Functions. ECRYPT hash workshop, 2007.
5. B. Bilgin, A. Bogdanov, M. Knezevic, F. Mendel, and Q. Wang. FIDES: Lightweight Authenticated Cipher with Side-Channel Resistance for Constrained Hardware. In *CHES*, volume 8086 of *Lecture Notes in Computer Science*, pages 142–158. Springer, 2013.
6. A. Bogdanov, F. Mendel, F. Regazzoni, V. Rijmen, and E. Tischhauser. ALE: AES-based lightweight authenticated encryption.
7. CAESAR. CAESAR: Competition for Authenticated Encryption: Security, Applicability,and Robustness. 2013.
8. G. Jakimoski and S. Khajuria. ASC-1: An Authenticated Encryption Stream Cipher. In *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 356–372. Springer, 2012.
9. R. S. Manjunath. Provably secure authenticated encryption modes. Masters Thesis, Indraprastha Institute of Information Technology, Delhi, 2013.
10. D. A. McGrew and J. Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In *INDOCRYPT*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.
11. D. Moody, S. Paul, and D. Smith-Tone. Improved Indifferentiability Security Bound for the JH Mode. In *3rd SHA-3 Candidate Conference*, 2012.
12. P. Rogaway, M. Bellare, and J. Black. OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption. *ACM Trans. Inf. Syst. Secur.*, 6(3):365–403, 2003.
13. M. J. O. Saarinen. CBEAM: Efficient Authenticated Encryption from Feebly One-Way ϕ Functions. In *CT-RSA*, volume 8366 of *Lecture Notes in Computer Science*, pages 251–269. Springer, 2014.
14. D. Whiting, N. Ferguson, and R. Housley. Counter with CBC-MAC (CCM). *Request for Comments (RFC)*, (3610), 2003.
15. H. Wu. The Hash Function JH. Submission to NIST (round 3), 2011.
16. H. Wu and B. Preneel. AEGIS: A Fast Authenticated Encryption Algorithm. In *Selected Areas in Cryptography*, 2013.

Table 4. Game G_0 perfectly simulates $(JHAE - \pi, \pi^{-1})$

```
Game $G_0$
Initialize:
$K \longleftarrow \{0,1\}^n$;
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
— on $O_1$ -query (N,A,M) —
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 0$ to $p - 1$ do:
    $y'_i \parallel y_i = O_2(x'_i \parallel x_i)$;
    $x'_{i+1} = y'_i \oplus m_{i+1}$;
    $x_{i+1} = y_i \oplus m_i$
end for
$y'_p \parallel y_p = O_2(x'_p \parallel x_p)$;
$x_{p+1} = y_p \oplus m_p$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
$T = x_{p+1} \oplus K$
Return $(C, T)$
— on $O_2$-query m—
$v = \pi(m)$
return $v$
— on $O_3$-query v—//Inverse Query
$m = \pi^{-1}(v)$
return $m$
```

Table 5. In game G_1 the permutations π and π^{-1} are simulated .

```
Game $G_1$
Initialize:
$X = \emptyset$ ;$K \longleftarrow \{0,1\}^n$;
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
— on $O_1$ -query (N,A,M) —
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 0$ to $p - 1$ do:
    $y'_i \parallel y_i = O_2(x'_i \parallel x_i)$;
    $x'_{i+1} = y'_i \oplus m_{i+1}$;
    $x_{i+1} = y_i \oplus m_i$
end for
$y'_p \parallel y_p = O_2(x'_p \parallel x_p)$;
$x_{p+1} = y_p \oplus m_p$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
$T = x_{p+1} \oplus K$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X$ then return $v$
else $v \longleftarrow \{0,1\}^{2n}$
if $\exists (m', v') \in X$ S.T $v' = v$ then
$v \leftarrow \{0,1\}^{2n} \backslash \{v' : (m', v') \in X\}$
$X = X \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X$ then return $m$
else $m \longleftarrow \{0,1\}^{2n}$
if $\exists (m', v') \in X$ S.T $m' = m$ then
$m \leftarrow \{0,1\}^{2n} \backslash \{m' : (m', v') \in X\}$
$X = X \cup (m, v)$
return $m$
```

Table 6. In game G_2 the bad event type-0 may occur.

```
Game $G_2$
Initialize:
$X = \emptyset$ ;$K \longleftarrow \{0,1\}^n$;
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
— on $O_1$ -query (N,A,M) —
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 0$ to $p - 1$ do:
    $y'_i \parallel y_i = O_2(x'_i \parallel x_i)$;
    $x'_{i+1} = y'_i \oplus m_{i+1}$;
    $x_{i+1} = y_i \oplus m_i$
end for
$y'_p \parallel y_p = O_2(x'_p \parallel x_p)$;
$x_{p+1} = y_p \oplus m_p$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
$T = x_{p+1} \oplus K$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X$ then return $v$
else $v \longleftarrow \{0,1\}^{2n}$
if $\exists(m', v') \in X$ S.T $v' = v$ then $bad_0 \leftarrow true$
$X = X \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X$ then return $m$
else $m \longleftarrow \{0,1\}^{2n}$
if $\exists(m', v') \in X$ S.T $m' = m$ then $bad_0 \leftarrow true$
$X = X \cup (m, v)$
return $m$
```

Table 7. In game G_3 oracle O_2 in oracle O_1 is simulated.

```
Game $G_3$
Initialize:
$X = \emptyset$ ;$K \longleftarrow \{0,1\}^n$;
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
— on $O_1$ -query (N,A,M) —
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 0$ to $p - 1$ do:
    if $(x'_i \parallel x_i, y'_i \parallel y_i) \in X$ then return $y'_i \parallel y_i$
    else $y'_i \parallel y_i \longleftarrow \{0,1\}^{2n}$
    if $\exists((x'_i \parallel x_i)', (y'_i \parallel y_i)') \in X$ S.T $(y'_i \parallel y_i)' = y'_i \parallel y_i$ then $bad_0 \leftarrow true$
     $X = X \cup (x'_i \parallel x_i, y'_i \parallel y_i)$
     $x'_{i+1} = y'_i \oplus m_{i+1}$;
     $x_{i+1} = y_i \oplus m_i$
end for
if $(x'_p \parallel x_p, y'_p \parallel y_p) \in X$ then return $y'_p \parallel y_p$
else $y'_p \parallel y_p \longleftarrow \{0,1\}^{2n}$
if $\exists((x'_p \parallel x_p)', (y'_p \parallel y_p)') \in X$ S.T $(y'_p \parallel y_p)' = y'_p \parallel y_p$ then $bad_0 \leftarrow true$
$X = X \cup (x'_p \parallel x_p, y'_p \parallel y_p)$
$x_{p+1} = y_p \oplus m_p$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
$T = x_{p+1} \oplus K$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X$ then return $v$
else $v \longleftarrow \{0,1\}^{2n}$
if $\exists(m', v') \in X$ S.T $v' = v$ then $bad_0 \leftarrow true$
$X = X \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X$ then return $m$
else $m \longleftarrow \{0,1\}^{2n}$
if $\exists(m', v') \in X$ S.T $m' = m$ then $bad_0 \leftarrow true$
$X = X \cup (m, v)$
return $m$
```

Table 8. In game G_4 bad event type-1 may occur.

```
Game $G_4$
Initialize:
$X_{O_1} = X_{O_2} = \emptyset$ ; $X = X_{O_1} \parallel X_{O_2}$; $K \longleftarrow \{0,1\}^n$;
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
— on $O_1$ -query (N,A,M) —
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 0$ to $p - 1$ do:
    if $(x'_i \parallel x_i, y'_i \parallel y_i) \in X_{O_1}$ then return $y'_i \parallel y_i$
     else if $(x'_i \parallel x_i, y'_i \parallel y_i) \in X_{O_2}$ then $bad_1 \leftarrow true$
    else $y'_i \parallel y_i \longleftarrow \{0,1\}^{2n}$
    if $\exists((x'_i \parallel x_i)', (y'_i \parallel y_i)') \in X$ S.T $(y'_i \parallel y_i)' = y'_i \parallel y_i$ then $bad_0 \leftarrow true$
     $X_{O_1} = X_{O_1} \cup (x'_i \parallel x_i, y'_i \parallel y_i)$
     $x'_{i+1} = y'_i \oplus m_{i+1}$;
     $x_{i+1} = y_i \oplus m_i$
end for
if $(x'_p \parallel x_p, y'_p \parallel y_p) \in X_{O_1}$ then return $y'_p \parallel y_p$
else if $(x'_p \parallel x_p, y'_p \parallel y_p) \in X_{O_2}$ then $bad_1 \leftarrow true$
else $y'_p \parallel y_p \longleftarrow \{0,1\}^{2n}$
if $\exists((x'_p \parallel x_p)', (y'_p \parallel y_p)') \in X$ S.T $(y'_p \parallel y_p)' = y'_p \parallel y_p$ then $bad_0 \leftarrow true$
$X_{O_1} = X_{O_1} \cup (x'_p \parallel x_p, y'_p \parallel y_p)$
$x_{p+1} = y_p \oplus m_p$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
$T = x_{p+1} \oplus K$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X$ then return $v$
else $v \longleftarrow \{0,1\}^{2n}$
if $\exists(m', v') \in X$ S.T $v' = v$ then $bad_0 \leftarrow true$
$X_{O_2} = X_{O_2} \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X$ then return $m$
else $m \longleftarrow \{0,1\}^{2n}$
if $\exists(m', v') \in X$ S.T $m' = m$ then $bad_0 \leftarrow true$
$X_{O_2} = X_{O_2} \cup (m, v)$
return $m$
```

Table 9. In G_5 , bad event type-2 may occur.

```
Game $G_5$
Initialize:
$X_{O_1} = X_{O_2} = W_{O_1} = W_{O_2} = Y_{O_1} = Y_{O_2} = \emptyset$ ; $X = X_{O_1} \parallel X_{O_2}$; $W = W_{O_1} \parallel W_{O_2}$; $Y = Y_{O_1} \parallel Y_{O_2}$;
$K \longleftarrow \{0,1\}^n$;
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
— on $O_1$ -query (N,A,M) —
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 0$ to $p - 1$ do:
    if $(x'_i \parallel x_i, y'_i \parallel y_i) \in X_{O_1}$ then return $y'_i \parallel y_i$
     else if $(x'_i \parallel x_i, y'_i \parallel y_i) \in X_{O_2}$ then $bad_1 \leftarrow true$
    else $y'_i \parallel y_i \longleftarrow \{0,1\}^{2n}$
    if $\exists((x'_i \parallel x_i)', (y'_i \parallel y_i)') \in X$ S.T $(y'_i \parallel y_i)' = y'_i \parallel y_i$ then $bad_0 \leftarrow true$
     $X_{O_1} = X_{O_1} \cup (x'_i \parallel x_i, y'_i \parallel y_i)$
     $W_{O_1} = W_{O_1} \cup (x'_i \parallel x_i)$, $Y_{O_1} = Y_{O_1} \cup (y'_i \parallel y_i)$
     $x'_{i+1} = y'_i \oplus m_{i+1}$;
     $x_{i+1} = y_i \oplus m_i$
end for
if $(x'_p \parallel x_p, y'_p \parallel y_p) \in X_{O_1}$ then return $y'_p \parallel y_p$
else if $(x'_p \parallel x_p, y'_p \parallel y_p) \in X_{O_2}$ then $bad_1 \leftarrow true$
else $y'_p \parallel y_p \longleftarrow \{0,1\}^{2n}$
if $\exists((x'_p \parallel x_p)', (y'_p \parallel y_p)') \in X$ S.T $(y'_p \parallel y_p)' = y'_p \parallel y_p$ then $bad_0 \leftarrow true$
$X_{O_1} = X_{O_1} \cup (x'_p \parallel x_p, y'_p \parallel y_p)$
$W_{O_1} = W_{O_1} \cup (x'_p \parallel x_p)$, $Y_{O_1} = Y_{O_1} \cup (y'_p \parallel y_p)$
$x_{p+1} = y_p \oplus m_p$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
$T = x_{p+1} \oplus K$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X_{O_2}$ then return $v$
if $m \in W_{O_1}$ then $bad_2 \leftarrow true$
else $v \longleftarrow \{0,1\}^{2n}$
if $\exists(m', v') \in X$ S.T $v' = v$ then $bad_1 \leftarrow true$
$X_{O_2} = X_{O_2} \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X_{O_2}$ then return $m$
if $v \in Y_{O_1}$ then $bad_2 \leftarrow true$
else $m \longleftarrow \{0,1\}^{2n}$
if $\exists(m', v') \in X$ S.T $m' = m$ then $bad_1 \leftarrow true$
$X_{O_2} = X_{O_2} \cup (m, v)$
return $m$
```

Table 10. In game G_6 O_1 does not keeps the history of intermediate queries.

```
Game $G_6$
Initialize:
$X = \emptyset$ ;$K \longleftarrow \{0,1\}^n$;
$IV = 0; m_0 = N$
$x'_0 = IV \oplus m_0$; $x_0 = K$
— on $O_1$ -query (N,A,M) —
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 0$ to $p - 1$ do:
    $y'_i \parallel y_i \leftarrow \{0,1\}^{2n}$;
    $x'_{i+1} = y'_i \oplus m_{i+1}$;
    $x_{i+1} = y_i \oplus m_i$
end for
$y'_p \parallel y_p \leftarrow \{0,1\}^{2n}$;
$x_{p+1} = y_p \oplus m_p$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
$T = x_{p+1} \oplus K$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X$ then return $v$
else $v \longleftarrow \{0,1\}^{2n}$
$X = X \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X$ then return $m$
else $m \longleftarrow \{0,1\}^{2n}$
$X = X \cup (m, v)$
return $m$
```

Table 11. In game G_7 , blocks of ciphertext and tag value are generated randomly.

```
Game $G_7$
Initialize:
$X = \emptyset$
— on $O_1$-query (N,A,M)—
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 1$ to $p$ do:
    $x'_i \longleftarrow \{0,1\}^n$
end for
$T \longleftarrow \{0,1\}^n$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X$ then return $v$
else $v \longleftarrow \{0,1\}^{2n}$
$X = X \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X$ then return $m$
else $m \longleftarrow \{0,1\}^{2n}$
$X = X \cup (m, v)$
return $m$
```

Table 12. In G_8 there is a switch of random permutation to random function.

```
Game $G_8$
Initialize:
$X = \emptyset$
— on $O_1$-query (N,A,M)—
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
for $i = 1$ to $p$ do:
    $x'_i \longleftarrow \{0,1\}^n$
end for
$T \longleftarrow \{0,1\}^n$
$C = x'_{l+1} \parallel x'_{l+2} \parallel ... \parallel x'_p$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X$ then return $v$
else $v \longleftarrow \{0,1\}^{2n}$
if $\exists (m', v') \in X$ S.T $v' = v$ then
$v \leftarrow \{0,1\}^{2n} \backslash \{v' : (m', v') \in X\}$
$X = X \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X$ then return $m$
else $m \longleftarrow \{0,1\}^{2n}$
if $\exists (m', v') \in X$ S.T $m' = m$ then
$m \leftarrow \{0,1\}^{2n} \backslash \{m' : (m', v') \in X\}$
$X = X \cup (m, v)$
return $m$
```

Table 13. Game G_9 perfectly simulates an ideal system.

```
Game $G_9$
Initialize:
$X = \emptyset$
— on $O_1$-query (N,A,M)—
$pad(A) \| pad(M) = m_1 \parallel m_2 \parallel ... \parallel m_p$
$C \longleftarrow \{0,1\}^{|Pad(M)|}$
$T \longleftarrow \{0,1\}^n$
Return $(C, T)$
— on $O_2$-query m—
if $(m, v) \in X$ then return $v$
else $v \longleftarrow \{0,1\}^{2n}$
if $\exists (m', v') \in X$ S.T $v' = v$ then
$v \leftarrow \{0,1\}^{2n} \backslash \{v' : (m', v') \in X\}$
$X = X \cup (m, v)$
return $v$
— on $O_3$-query v—//Inverse Query
if $(m, v) \in X$ then return $m$
else $m \longleftarrow \{0,1\}^{2n}$
if $\exists (m', v') \in X$ S.T $m' = m$ then
$m \leftarrow \{0,1\}^{2n} \backslash \{m' : (m', v') \in X\}$
$X = X \cup (m, v)$
return $m$
```