Attacks on the Authenticated Encryption Mode of Operation PAE

Debrup Chakraborty and Mridul Nandi

***Abstract*—We show several concrete attacks on an authenticated encryption scheme PAE which appeared in IEEE Transactions on Information Theory, Vol. 56, no. 8, pp. 4025–4037. Additionally we show some flaws and oversights in the analysis (presented in the same paper) used to prove PAE to be a secure authenticated encryption scheme.**

***Index Terms*—Symmetric Cryptography, Block cipher mode of operation, Authenticated encryption, Forgery attack, Distinguishing attack.**

I. Introduction

Authenticated encryption (AE) schemes provide security in terms of both privacy and authentication. These schemes have huge practical applications and there are several constructions of AE schemes available in the literature. A current competition of authenticated encryption schemes is considering more than fifty different proposals [2].

In [6] a new single pass AE scheme named PAE was proposed. It was claimed that PAE enjoys numerous benefits in terms of efficiency and usability and in terms of efficiency it compares well with the famous OCB mode [5], [4]. In addition to PAE, [6] describes PAE-1, which is a variant of PAE and also describes PAEAD, PAEAD-1 which are schemes for authenticated encryption with associated data (AEAD). PAEAD and PAEAD-1 are built over PAE and PAE-1, respectively.

In this paper we show several concrete attacks on PAE which proves that PAE is not a secure AE. The attacks can also be extended to PAE-1, PAEAD and PAEAD-1. As would be evident from the rest of the paper, the attacks though simple are easy to overlook.

In [6], PAE was proven to be secure, thus the attacks prove that the security theorem regarding PAE is false. We also try to investigate the oversights that led the author to prove a false theorem.

A novel technique for analysing/proving security of AE schemes was proposed in [6]. Informally the technique consist of analyzing the security and authenticity of the cipher separately and further combining them to argue regarding the AE-security of the scheme. The crux of this technique is summarized as Proposition 6 in [6] (see Eq. (6) in this paper). Our analysis reveals that this proposition is false. We revisit the proof of the proposition presented in [6] and point out where the argument fails. Finally, we construct an (insecure) AE scheme called insOCB, which serves as a counterexample to a more general version of Proposition 6 in [6]. Through insOCB we can specifically point out why the general technique proposed in [6] fails.

Notations. For binary strings X, Y, $X||Y$ denotes the concatenation of X and Y. $|X|$ denotes the length of X in bits. For a binary string X, $\mathsf{Format}(X, n)$ returns $(X_1, X_2, \ldots, X_m)$, where $m = \lceil |X|/n \rceil$, and $|X_i| = n$, for $1 \leq i \leq m-1$ and $1 \leq |X_m| \leq n$. For a positive integer $i \leq 2^n - 1$, $\mathsf{bin}_n(i)$ denotes the n-bit binary representation of i. For a binary string X, where $|X| \geq r$, $\mathsf{First}_r(X)$ denotes the first r bits of X. $x \xleftarrow{\$} \mathcal{S}$, would denote x to be an uniform random element in the finite set $\mathcal{S}$. By $\mathsf{Perm}(n)$ we will denote the set of all permutations on the set $\{0,1\}^n$. By $\mathbb{F}_q$ we denote a finite field with q elements.

II. The Scheme PAE from [6]

The scheme PAE uses an n-bit block cipher E_K to map a pair (N, P) to $(C; \mathsf{tag})$, where N is the nonce, P is the plaintext and $(C; \mathsf{tag})$ is the ciphertext. The encryption and decryption procedures of PAE are shown in Figure 1. The schematic diagram in Figure 2 shows the encryption for a four block message.

In Fig. 1, the mask $\Gamma_{\gamma,i}$ is defined to be $\psi^i(\gamma)$ where ψ is a (publicly known) linear map from $\mathbb{F}_{2^n}$ to itself whose minimal polynomial over $\mathbb{F}_2$ is primitive and of degree n. For the attacks, the actual nature of the map is not of consequence, hence we do not discuss that here.

III. The Attacks

First we informally discuss the security notions for an AE scheme. To appreciate the attacks this informal description would be enough.

An AE scheme f is considered to be secure if it satisfies the following conditions.

1) **Privacy:** The outputs of f are indistinguishable from random strings to a computationally bounded adversary which can see the outputs of f for messages and nonces of its choice.
   For defining privacy, an adversary is given an oracle. The oracle can either be the encryption algorithm of the AE scheme or it can just return random strings. We call these oracles as *real* and *random*, respectively.

Debrup Chakraborty is with the Computer Science Department, CINVESTAV-IPN, 2508 Av. IPN, San Pedro Zacatenco, Mexico City 07360, Mexico. Email: debrup@cs.cinvestav.mx

Mridul Nandi is with the Applied Statistics Unit, Indian Statistical Institute, 203 B.T. Road, Kolkata, India 700108. Email: mridul@isical.ac.in

$\mathsf{PAE.Encrypt}_K(N, P)$:	$\mathsf{PAE.Decrypt}_K(N, C; \mathsf{tag})$:
1. $(P_1, \ldots, P_m) \leftarrow \mathsf{Format}(P, n)$;	1. $(C_1, \ldots, C_m) \leftarrow \mathsf{Format}(C, n)$;
2. $\gamma \leftarrow E_K^{-1}(N)$;	2. $\gamma \leftarrow E_K^{-1}(N)$;
3. **for** $1 \le i \le m-1$,	3. **for** $1 \le i \le m-1$,
4. $C_i \leftarrow E_K(P_i \oplus \Gamma_{\gamma,i}) \oplus \Gamma_{\gamma,i}$;	4. $P_i \leftarrow E_K^{-1}(C_i \oplus \Gamma_{\gamma,i}) \oplus \Gamma_{\gamma,i}$;
5. **end for**	5. **end for**
6. **if** $\lvert P_m \rvert = n$ then	6. **if** $\lvert C_m \rvert = n$ then
7. $C_m \leftarrow E_K(P_m \oplus \Gamma_{\gamma,m}) \oplus \Gamma_{\gamma,m}$;	7. $P_m \leftarrow E_K^{-1}(C_m \oplus \Gamma_{\gamma,m}) \oplus \Gamma_{\gamma,m}$;
8. $\mathsf{sum} \leftarrow P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m$;	8. $\mathsf{sum}' \leftarrow P_1 \oplus \cdots \oplus P_{m-1} \oplus C_m$;
9. **else**	9. **else**
10. $r \leftarrow \lvert P_m \rvert$; $P_m \leftarrow P_m \lvert\lvert 10^{n-r-1}$;	10. $r \leftarrow \lvert C_m \rvert$; $T_m \leftarrow C_m \lvert\lvert 10^{n-r-1}$;
11. $\mathsf{tmp} \leftarrow E_K^{-1}(\mathsf{bin}_n(r) \oplus \Gamma_{\gamma,m})$;	11. $\mathsf{tmp} \leftarrow E_K^{-1}(\mathsf{bin}_n(r) \oplus \Gamma_{\gamma,m})$;
12. $C_m \leftarrow \mathsf{First}_r(P_m \oplus \mathsf{tmp})$; $T_m = C_m \lvert\lvert (10^{n-r-1})$;	12. $P_m \leftarrow \mathsf{First}_r(T_m \oplus \mathsf{tmp})$;
13. $\mathsf{sum} \leftarrow P_1 \oplus \cdots \oplus P_{m-1} \oplus T_m \oplus \Gamma_{\gamma,m+1}$;	13. $\mathsf{sum}' \leftarrow P_1 \oplus \cdots \oplus P_{m-1} \oplus T_m \oplus \Gamma_{\gamma,m+1}$;
14. **end if**;	14. **end if**;
15. **if** $(m = 1)$ then	15. **if** $(m = 1)$ then
16. $\delta \leftarrow E_K^{-1}(\gamma)$;	16. $\delta \leftarrow E_K^{-1}(\gamma)$;
17. $\mathsf{sum} \leftarrow \mathsf{sum} \oplus \delta$;	17. $\mathsf{sum}' \leftarrow \mathsf{sum}' \oplus \delta$;
18. **end if**;	18. **end if**;
19. $\mathsf{tag} \leftarrow E_K^{-1}(\mathsf{sum})$;	19. $\mathsf{tag}' \leftarrow E_K^{-1}(\mathsf{sum}')$;
20. **return** $(C_1 \lvert\lvert \cdots \lvert\lvert C_{m-1} \lvert\lvert C_m; \mathsf{tag})$;	20. **if** $\mathsf{tag} = \mathsf{tag}'$, **return** $P_1 \lvert\lvert P_2 \lvert\lvert \cdots \lvert\lvert P_m$,
	21. **else return** $\perp$;

Fig. 1. Encryption and decryption using PAE

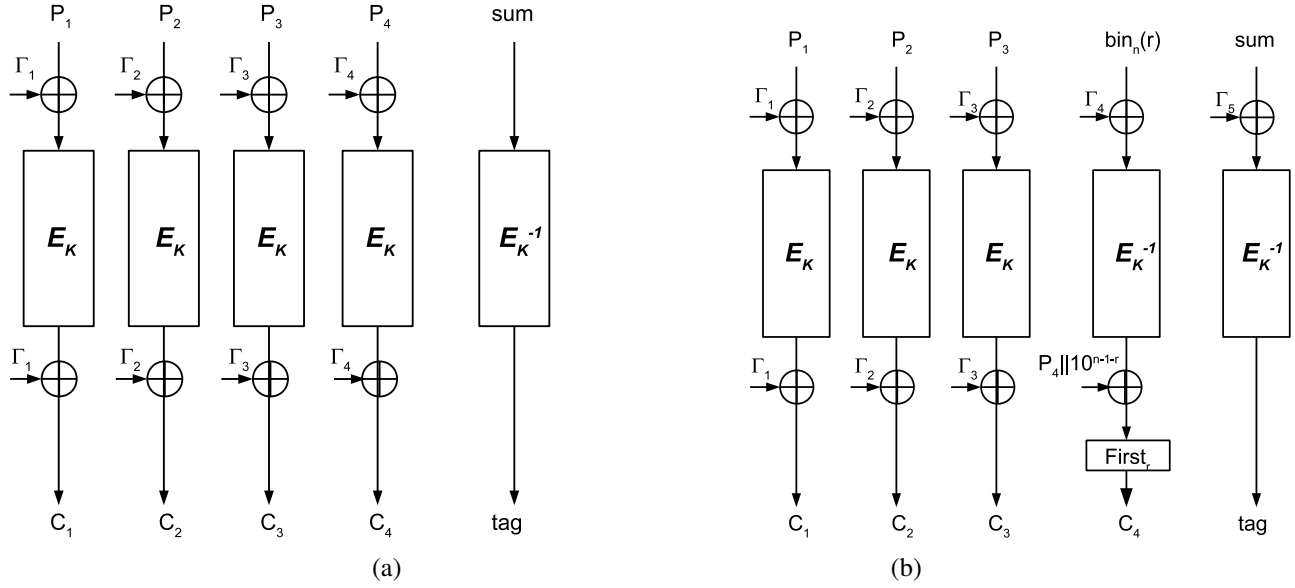

Fig. 2. Schematic diagram showing encryption of a four block message $P_1||P_2||P_3||P_4$ using PAE. (a) All blocks are n-bits long; $\mathsf{sum} = P_1 \oplus P_2 \oplus P_3 \oplus C_4$. (b) P_1, P_2, P_3 are n-bits long and $0 < |P_4| < n$; $\mathsf{sum} = P_1 \oplus P_2 \oplus P_3 \oplus C_4 || 10^{n-r+1}$.

The adversary can query the oracle; each query is of the form (N, P) where N is the nonce and P the plaintext. For each query the oracle returns $(C; \mathsf{tag})$, if the oracle is the real one then $(C; \mathsf{tag})$ corresponds to the real ciphertext otherwise it is a random string of appropriate length. The goal of the adversary is to detect whether the oracle is the *real* or the *random* one. The only restriction that the adversary has is that it cannot repeat nonces in its queries. If there exists an adversary which can distinguish with "high" probability by asking a "reasonable" number of queries in a "reasonable" amount of time then the AE scheme is considered to be insecure in terms of privacy.

2) **Authenticity:** An AE scheme is said to have the property of authenticity if it is difficult for any computationally bounded adversary to create a ciphertext which it has not seen before and which decrypts.
To define this property, an adversary is given oracle access to the encryption algorithm and allowed to see ciphertexts corresponding to nonce, message pairs of its

choice. Finally the adversary outputs a *forgery*, which consists of a nonce ciphertext pair $(N, (C; \mathsf{tag}))$. The forgery $(N, (C; \mathsf{tag}))$ is considered valid if the adversary has not obtained $(C; \mathsf{tag})$ in any of its previous queries (N, P). If $(C; \mathsf{tag})$ decrypts with "high probability", i.e., the decryption algorithm does not return $\perp$ then we say that the forgery is successful and the adversary has broken the scheme in terms of authenticity.

For formal definitions of these notions see [1], [5], [6]. We also briefly discuss in Section IV the formalism used in [6].

We present attacks against both privacy and authenticity of PAE, the attacks assume the block cipher E_K to be a random permutation, and additionally no specific property of the mapping ψ is used.

A. Attack on the authenticity of PAE for messages with partial last block

In this attack the adversary asks only a single query before it produces the forgery. The encryption query is (N, P), where N is an arbitrary nonce, and $P = P_1||P_2||P_3||P_4$, where $P_1, P_2, P_3 \in \{0,1\}^n$ and $P_4 = 0^{n-1}$. Additionally, $P_1 \oplus P_2 = \mathsf{bin}_n(n-1) \oplus 0^{n-1}1$. We assume that in response to this encryption query the adversary gets $(C; \mathsf{tag})$, where $C = C_1||C_2||C_3||C_4$, where $C_1, C_2, C_3 \in \{0,1\}^n$ and $C_4 \in \{0,1\}^{n-1}$.

The adversary finally outputs a forgery $(N, C_1||C_2||0^{n-1}; \mathsf{tag}'')$ where $\mathsf{tag}'' = C_4||0$.

First, note that this is a valid forgery as the ciphertext is not same as the ciphertext that it has seen before. Now we analyze that this ciphertext gets decrypted with high probability. Notice that, according to the encryption algorithm

$$\begin{aligned} C_4 &= P_4 \oplus \mathsf{First}_{n-1}(E_K^{-1}(\mathsf{bin}_n(n-1) \oplus \Gamma_{\gamma,4})) \\ &= \mathsf{First}_{n-1}(E_K^{-1}(\mathsf{bin}_n(n-1) \oplus \Gamma_{\gamma,4})). \end{aligned} \quad (1)$$

Thus, when the forged ciphertext $(N, C_1||C_2||0^{n-1}; \mathsf{tag}'')$ is decrypted we have the internal variable sum' as

$$\mathsf{sum}' = P_1 \oplus P_2 \oplus (0^{n-1}1) \oplus \Gamma_{\gamma,4} = \mathsf{bin}_n(n-1) \oplus \Gamma_{\gamma,4},$$

and the tag generated by the decryption algorithm is

$$\mathsf{tag}' = E_K^{-1}(\mathsf{sum}') = E_K^{-1}(\mathsf{bin}_n(n-1) \oplus \Gamma_{\gamma,4}). \quad (2)$$

From Eqs. (1) and (2), we see that $\mathsf{First}_{n-1}(\mathsf{tag}') = C_4$, and recall that the forged tag produced by the adversary is $\mathsf{tag}'' = C_4||0$. Thus, if the last bit of tag' is 0, then $\mathsf{tag}'' = \mathsf{tag}'$ and the forgery is successful. Assuming $E_K()$ to be a random permutation, the last bit of tag' is 0 with probability $1/2$ and so the forgery will be successful with probability $1/2$.

B. Attack on the authenticity of PAE for full block messages

The adversary asks a single encryption query $(N, P_1||P_2||0^n||P_4)$, where N is an arbitrary nonce and P_1, P_2, P_4 are arbitrary n-bit strings. As a response to this query, the adversary gets $(C_1||C_2||C_3||C_4; \mathsf{tag})$. Finally, the adversary outputs the forgery $(N, C_1||C_2||C_4; \mathsf{tag})$. Note, here the forged tag is same as that obtained as a response to the encryption query, but the ciphertext is different, thus this forgery is a valid one. Now, we see why this would always be a successful forgery.

Note, for the encryption query we have

$$\mathsf{sum} = P_1 \oplus P_2 \oplus 0^n \oplus C_4 = P_1 \oplus P_2 \oplus C_4,$$

thus,

$$\mathsf{tag} = E_K^{-1}(\mathsf{sum}) = E_K^{-1}(P_1 \oplus P_2 \oplus C_4). \quad (3)$$

While decrypting $(N, C_1||C_2||C_4; \mathsf{tag})$ the variable sum' in the decryption algorithm is

$$\mathsf{sum}' = P_1 \oplus P_2 \oplus C_4,$$

and

$$\mathsf{tag}' = E_K^{-1}(\mathsf{sum}') = E_K^{-1}(P_1 \oplus P_2 \oplus C_4). \quad (4)$$

From Equations (3) and (4) it is evident that the forgery is always successful.

C. Attack on the privacy of PAE

We design an adversary who distinguish the outputs of PAE from random strings. The adversary makes two encryption queries. The first encryption query is $(N, 0^n||0^n)$, and this query returns $(C_1||C_2; \mathsf{tag})$. Here, according to the encryption algorithm $\mathsf{sum} = C_2$ and $\mathsf{tag} = E_K^{-1}(C_2)$.

The second encryption query is $(C_2, \psi(\mathsf{tag})||\psi^2(\mathsf{tag}))$, this query returns $(CC_1||CC_2; \mathsf{tag}^*)$. Note, for this query, C_2 is the nonce and tag is the tag obtained in the previous query; the two plain text blocks are $\psi(\mathsf{tag})$ and $\psi^2(\mathsf{tag})$. As ψ is a publicly known map, hence it is possible for the adversary to compute $\psi(\mathsf{tag})$ and $\psi^2(\mathsf{tag}) = \psi(\psi(\mathsf{tag}))$.

After these two queries, the adversary follows the following procedure to distinguish:

if $CC_1 \oplus \psi(\mathsf{tag}) = CC_2 \oplus \psi^2(\mathsf{tag})$,
  return *real*;
else return *random*;

For the second query, according to the encryption algorithm,

$$\gamma = E_K^{-1}(C_2) = \mathsf{tag}.$$

Thus, we have

$$\begin{aligned} CC_1 &= E_K(\psi(\mathsf{tag}) \oplus \psi(\mathsf{tag})) \oplus \psi(\mathsf{tag}) \\ &= E_K(0^n) \oplus \psi(\mathsf{tag}); \\ CC_2 &= E_K(\psi^2(\mathsf{tag}) \oplus \psi^2(\mathsf{tag})) \oplus \psi^2(\mathsf{tag}) \\ &= E_K(0^n) + \psi^2(\mathsf{tag}). \end{aligned}$$

Thus if the oracle is the real one then with probability 1,

$$CC_1 \oplus \psi(\mathsf{tag}) = CC_2 \oplus \psi^2(\mathsf{tag}). \quad (5)$$

In case the oracle is random, then CC_1 and CC_2 are uniform random strings from $\{0,1\}^n$, and (5) is true with probability $1/2^n$. Thus this adversary is correct with probability $1 - 1/2^n$.

D. Other Schemes in [6]

In addition to PAE, [6] describes the following schemes:

1) A message authentication code (MAC) named iPMAC and some variants of it.
2) An AE scheme called PAE-1 which is a modification of PAE.
3) Authenticated encryption with associated data (AEAD) schemes PAEAD and PAEAD-1.

We discuss next the effect of our attacks on these schemes.

1) iPMAC is a block cipher based parallel MAC whose functionality and security goals are different from PAE. Our attacks *does not seem to violate* the claimed security of iPMAC and its variants.
2) PAE-1 is derived by modifying PAE. In the encryption algorithm for PAE (see Figure 1) if all the inverse calls to the block cipher, which occurs in lines 2, 11, 16 and 19, are replaced by $E_K()$, then we obtain PAE-1. It is easy to verify that all the attacks described in Section III can also be applied to PAE-1.
3) The PAEAD construction uses PAE and iPMAC. An AEAD scheme takes as input (N, H, P), where N is the nonce, H the header (associated data) and P the plaintext. The goal is to authenticate both H and P, but to only encrypt P. For an input (N, H, P) encryption algorithm for PAEAD computes $(C; \mathsf{tag}_1) = \mathsf{PAE.Encrypt}_K(N, P)$, additionally it computes tag_2, which is the message authentication code for H using iPMAC, finally it outputs $(C; \mathsf{tag}_1 \oplus \mathsf{tag}_2)$. For PAEAD-1, instead of PAE.Encrypt, PAE-1.Encrypt is used.
   Note that both the PAEAD constructions allow empty headers, and when the header is empty then PAEAD gives the same output as PAE. This signifies that all the attacks on PAE described in Section III are applicable to PAEAD also. Moreover, PAEAD can also be attacked by using queries containing non-empty headers. In particular the attack in Section III-B directly applies to PAEAD. The adversary asks a query $(N, H, P_1||P_2||0^n||P_4)$, where H is an arbitrary string. It gets back $(C_1||C_2||C_3||C_4; \mathsf{tag})$ as the response; and finally outputs $(N, H, C_1||C_2||C_4; \mathsf{tag})$ as the forgery. Following the explanations in Section III-B it can be easily verified that irrespective of the structure of iPMAC, this forgery is always successful for PAEAD.

IV. Oversights in the Original Analysis

The original analysis of PAE depends on (an attempt to obtain) a new security characterization of AE. We first discuss the techniques and results used in [6] for the analysis.

Let $\mathcal{N}$, $\mathcal{X}$, $\mathcal{Y}$ be finite nonempty sets of binary strings. $\mathcal{G}$ be a set of functions mapping $\mathcal{X}$ to $\mathcal{Y}$. Let g be a uniform random function from $\mathcal{G}$. The authenticity of g is defined as follows. An adversary $\mathcal{A}$ has access to g as an oracle and can submit queries in an adaptive manner, i.e., for $1 \leq i \leq q-1$ each query $x^{(i)}$ of $\mathcal{A}$ is answered by $y^{(i)} = g(x^{(i)})$. Finally, $\mathcal{A}$ outputs a "forged" pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and is said to be successful if $y = g(x)$. The pair (x, y) must not be equal to any previous pair $(x^{(i)}, y^{(i)})$, for $1 \leq i \leq q-1$. Let this event of $\mathcal{A}$ being successful be denoted by $\mathsf{Succ}(A)$. The advantage of $\mathcal{A}$ in breaking the authenticity of g is defined as

$$\mathbf{Adv}_g^{\text{auth}}(\mathcal{A}) = \Pr[\mathsf{Succ}(\mathcal{A})].$$

A (q, σ)-adversary is an adversary which makes at most q queries with query complexity at most σ. Define $\mathbf{Adv}_g^{\text{auth}}(q, \sigma)$ to be the maximum of $\mathbf{Adv}_g^{\text{auth}}(\mathcal{A})$, taken over all (q, σ)-adversaries $\mathcal{A}$.

Let $\mathcal{F}_n[\mathcal{N}, \mathcal{X}]$ be a set of functions $f : \mathcal{N} \times \mathcal{X} \to \mathcal{X} \times \{0,1\}^n$ such that if $f(N, X) = (Y; \mathsf{tag})$, then $|X| = |Y|$. The following notions for f were defined in [6].

1) $f^{\text{main}} : \mathcal{N} \times \mathcal{X} \to \mathcal{X}$ is defined to be $f^{\text{main}}(N, X) = Y$ if $f(N, X) = (Y, \mathsf{tag})$.
2) The function f is said to be an AE function, if for every $N \in \mathcal{N}$, $f_N^{\text{main}}(\cdot) \stackrel{\triangle}{=} f^{\text{main}}(N, .)$ is a length preserving permutation.
3) Let f be an AE function, then $\widetilde{f} : \mathcal{N} \times \mathcal{X} \to \{0,1\}^n$ is defined as $\widetilde{f}(N, Y) = \mathsf{tag}$ if $f(N, X) = (Y; \mathsf{tag})$, for some $X \in \mathcal{X}$. $\widetilde{f}$ is called the authentication function associated with f.

Security of an AE function is defined in terms of privacy and authenticity. Let f be a uniform random AE function from $\mathcal{F}_n[\mathcal{N}, \mathcal{X}]$. For defining privacy of f, $\mathcal{A}$ is allowed to query f, i.e., for $1 \leq i \leq q$, $\mathcal{A}$ queries $(N^{(i)}, X^{(i)}) \in \mathcal{N} \times \mathcal{X}$ and gets back $(Y^{(i)}; \mathsf{tag}^{(i)})$. The only restriction is that no two queries can have the same nonce $N^{(i)}$. Finally $\mathcal{A}$ outputs a bit. $\mathcal{A}^f \Rightarrow b$, denotes the event that $\mathcal{A}$ interacting with the oracle f outputs the bit b. The privacy advantage of $\mathcal{A}$ in breaking f is defined as

$$\mathbf{Adv}_f^{\text{priv}}(\mathcal{A}) = \Pr[\mathcal{A}^{f(.,.)} \Rightarrow 1] - \Pr[\mathcal{A}^{\$(.,.)} \Rightarrow 1],$$

where $\$(.,.)$ on input (N, X) returns a random binary string of length $|X| + n$. $\mathbf{Adv}_f^{\text{priv}}(q, \sigma)$ is defined to be the maximum of $\mathbf{Adv}_f^{\text{priv}}(\mathcal{A})$ taken over all (q, σ)-adversaries $\mathcal{A}$.

For an AE function f, the privacy of f^{main} is defined in the same way, except that for the oracle queries the adversary does not get the tag as the part of the response.

For defining authenticity of an AE function f, an adversary $\mathcal{A}$ is given oracle access to f and can adaptively query $(N^{(i)}, X^{(i)})$ to get back $(Y^{(i)}; \mathsf{tag}^{(i)})$ as response. As before, $\mathcal{A}$ is not allowed to repeat a nonce. Suppose $q-1$ such queries are made. Finally, $\mathcal{A}$ outputs a forgery $(N^{(q)}, Y^{(q)}; \mathsf{tag}^{(q)})$. For the forgery to be valid, it is required that $(N^{(q)}, Y^{(q)}; \mathsf{tag}^{(q)}) \neq (N^{(i)}, Y^{(i)}; \mathsf{tag}^{(i)})$, for $1 \leq i \leq q-1$. There is no restriction on $N^{(q)}$, i.e., it can be equal to some $N^{(i)}$, $1 \leq i \leq q-1$. The forgery is said to be successful if there exists a $X \in \mathcal{X}$, such that $f(N^{(q)}, X) = (Y^{(i)}; \mathsf{tag}^{(q)})$. Let the event that $\mathcal{A}$ commits a successful forgery be $\mathsf{Suc}(\mathcal{A})$, then define the advantage of $\mathcal{A}$ in breaking the authenticity of f as

$$\mathbf{Adv}_f^{\text{ae-auth}}(\mathcal{A}) = \Pr[\mathsf{Suc}(\mathcal{A})].$$

Define $\mathbf{Adv}_f^{\text{ae-auth}}(q, \sigma)$ as the maximum of $\mathbf{Adv}_f^{\text{ae-auth}}(\mathcal{A})$, where the maximum is taken over all (q, σ)-adversaries $\mathcal{A}$.

A. The central flaw in the analysis

The main tool that [6] uses to prove security of PAE is a (supposed) relationship with AE security of an AE function f with the privacy and authenticity of f^{main} and $\tilde{f}$. Proposition 6 in [6], states the following:
Given an AE function f,

$$\mathbf{Adv}^{\text{ae-auth}}_{f}(q,\sigma) \leq \mathbf{Adv}^{\text{priv}}_{f^{\text{main}}}(q,\sigma) + \mathbf{Adv}^{\text{auth}}_{\tilde{f}}(q,\sigma). \quad (6)$$

This proposition reduces the task of proving f to be a secure AE to two different tasks of proving f^{main} to be secure in terms of privacy and $\tilde{f}$ to be secure in terms of authentication. This may be very useful for proving security of many AE schemes, but unfortunately this proposition is incorrect.

PAE is itself a counterexample to this proposition. As claimed in Theorem 7 of [6], PAE^{main} is indeed secure in terms of privacy. Also $\widetilde{\text{PAE}}$ is secure in terms of authenticity (as claimed in Proposition 8 of [6]), but PAE is not a secure AE as demonstrated by the attacks in Section III.

Note that Theorem 7 in [6] claims both PAE and PAE^{main} to be secure in terms of privacy. But, PAE is not secure in terms of privacy as demonstrated by the attack in Section III-C. The specific attack on the privacy of PAE does not violate the privacy of PAE^{main}, as the attack does not work if the adversary does not get to see the tags.

Though the counter example of PAE is enough to invalidate (6), still it is worthwhile to take a closer look at the proof for (6) in [6] and investigate the flaw in the argument. First we present (almost verbatim) the proof presented in [6].

The proof of (6) in [6]: Let f be an AE function and $\mathcal{A}$ be any arbitrary (q,σ)-adversary attacking the authenticity of $\mathcal{A}$. $\mathcal{A}$ asks $(q-1)$ valid queries to its oracle (which is f), where each query is of the form $(N^{(i)}, P^{(i)})$, $1 \leq i \leq q-1$. For each query it gets $(C^{(i)}; \mathsf{tag}^{(i)})$ as a response. Finally $\mathcal{A}$ outputs the forgery $(N^{(q)}, C^{(q)}; \mathsf{tag}^{(q)})$. Let $\mathsf{Succ}(\mathcal{A})$ be the event that the forgery is successful, and let $p_0 = \Pr[\mathsf{Succ}(\mathcal{A})]$.

Let $\mathcal{B}$ be an adversary which tries to break the authenticity of $\tilde{f}$. $\mathcal{B}$ thus have $\tilde{f}$ as its oracle, and it runs $\mathcal{A}$ by answering to its queries as follows: For a query $(N^{(i)}, P^{(i)})$, $\mathcal{B}$ selects a uniform random string $Y^{(i)}$ such that $|Y^{(i)}| = |P^{(i)}|$, and queries its oracle with $(N^{(i)}, Y^{(i)})$; it gets as response $\mathsf{tag}^{(i)}$ and returns $(Y^i; \mathsf{tag}^{(i)})$ to $\mathcal{A}$. Finally $\mathcal{A}$ outputs a forgery and $\mathcal{B}$ outputs the same forgery. Let $\mathsf{Succ}(\mathcal{B})$ be the event that $\mathcal{B}$ is successful, and let $p_1 = \Pr[\mathsf{Succ}(\mathcal{B})]$.

Note that, in the above simulation, $\mathcal{B}$ is only successful when $\mathcal{A}$ is successful. The difference between p_0 and p_1 is that $Y^{(i)}$ is uniformly random instead of being the output of f on $(N^{(i)}, X^{(i)})$. This shows

$$p_0 - p_1 \leq \mathbf{Adv}^{\text{priv}}_{f^{\text{main}}}(q,\sigma). \quad (7)$$

Thus,

$$\begin{aligned}
\mathbf{Adv}^{\text{ae-auth}}_{f}(q,\sigma) &= p_0 \\
&= (p_0 - p_1) + p_1 \\
&\leq \mathbf{Adv}^{\text{priv}}_{f^{\text{main}}}(q,\sigma) + \Pr[\mathsf{Succ}(\mathcal{B})] \\
&\leq \mathbf{Adv}^{\text{priv}}_{f^{\text{main}}}(q,\sigma) + \mathbf{Adv}^{\text{auth}}_{\tilde{f}}(q,\sigma),
\end{aligned}$$

as desired ■

The main flaw in the argument occurs in (7), which can be written as $p_0 \leq p_1 + \mathbf{Adv}^{\text{priv}}_{f^{\text{main}}}(q,\sigma)$. Thus, to justify (7) we need to construct a reduction of the following type:

> Given a AE-forgery adversary $\mathcal{A}$ against an AE f there exists a forgery adversary $\mathcal{B}$ against $\tilde{f}$ and a privacy adversary $\mathcal{C}$ against f^{main}, such that
>
> $$\mathbf{Adv}^{\text{ae-auth}}_{f}(\mathcal{A}) \leq \mathbf{Adv}^{\text{auth}}_{f}(\mathcal{B}) + \mathbf{Adv}^{\text{priv}}_{f^{\text{main}}}(\mathcal{C}).$$

But, unfortunately this reduction is not explicitly constructed. It seems that there is no way to construct such a reduction. The problems that one encounters while constructing such a reduction are as follows:

1) $\mathcal{C}$, even with its real oracle is unable to provide the correct environment to either $\mathcal{A}$ or $\mathcal{B}$.
2) $\mathcal{C}$ has no way to evaluate whether a forgery is correct or not.

B. Another Counterexample

Here we provide another simple counterexample for a more general version of (6):
Given an AE function f,

$$\mathbf{Adv}^{\text{ae-auth}}_{f}(q,\sigma) \leq \mathbf{Adv}^{\text{priv}}_{f}(q,\sigma) + \mathbf{Adv}^{\text{auth}}_{\tilde{f}}(q,\sigma). \quad (8)$$

The above equation is a more general version of (6) as f being secure in terms of privacy implies f^{main} being secure in terms of privacy. This too is not true.

Note that PAE does not work as a counterexample for (8) as PAE is not secure both in terms of privacy and authentication. We describe an AE scheme which we call insOCB which is secure in terms of privacy and $\widetilde{\text{insOCB}}$ is secure in terms of authentication but insOCB is not secure in terms of authentication.

Let $\mathbb{I} = \{0, 1, \cdots, L\} \times \{0,1\}^n$, where L is an arbitrary large integer. Let $\mathbb{P} = \{\pi^{i,N} : (i,N) \in \mathbb{I}\}$ where each $\pi^{i,N}$ is an independent uniform random permutation drawn from $\mathsf{Perm}(n)$.

The message space $\mathcal{M}$ of insOCB contains binary strings whose lengths are multiples of n and are at most Ln bits long. The nonce space is $\{0,1\}^n$. The key for insOCB is $\mathbb{P}$, i.e., the key consists of $2^n(L+1)$ many independent and uniform random permutations. The encryption and decryption procedures for insOCB are shown in Figure 3.

The description of $\widetilde{\text{insOCB}}$, is basically the decryption algorithm in Figure 3. $\widetilde{\text{insOCB}}$ receives as input the key $\mathbb{P}$ and a pair (N, C). It follows the same procedure as insOCB.Decrypt, from line 1 to line 7, and finally outputs tag'.

Proposition 1: Let $\mathcal{A}$ be an arbitrary adversary, then

$$\begin{aligned}
\mathbf{Adv}^{\text{priv}}_{\text{insOCB}}(\mathcal{A}) &= 0, & (9)\\
\mathbf{Adv}^{\text{auth}}_{\widetilde{\text{insOCB}}}(\mathcal{A}) &\leq \frac{1}{2^n - 1}. & (10)
\end{aligned}$$

Proof: The privacy proof is immediate. Note that the adversary always queries with a different nonce N, hence after its queries it gets to see the image of only one point of each

$\mathsf{insOCB.Encrypt}_{\mathbb{P}}(N, P)$:	$\mathsf{insOCB.Decrypt}_{\mathbb{P}}(N, C; \mathsf{tag})$:
1. $(P_1, \ldots, P_m) \leftarrow \mathsf{Format}(P, n)$;	1. $(C_1, \ldots, C_m) \leftarrow \mathsf{Format}(C, n)$;
2. $\mathsf{sum} \leftarrow 0^n$	2. $\mathsf{sum}' \leftarrow 0^n$;
3. **for** $1 \le i \le m$,	3. **for** $1 \le i \le m$,
4. $C_i \leftarrow \pi^{i,N}(P_i)$;	4. $P_i \leftarrow \left(\pi^{i,N}\right)^{-1}(C_i)$;
5. $\mathsf{sum} \leftarrow \mathsf{sum} \oplus P_i$;	5. $\mathsf{sum}' \leftarrow \mathsf{sum}' \oplus P_i$;
6. **end for**	6. **end for**
7. $\mathsf{tag} \leftarrow \pi^{0,N}(\mathsf{sum})$;	7. $\mathsf{tag}' \leftarrow \pi^{0,N}(\mathsf{sum}')$;
8. **return** $(C_1 \| \cdots \| C_{m-1} \| C_m; \mathsf{tag})$;	8. **if** $\mathsf{tag} = \mathsf{tag}'$, **return** $P_1 \| P_2 \| \cdots \| P_m$,
	9. **else return** $\perp$;

Fig. 3. Encryption and decryption using insOCB

permutation involved in the queries. The image of a single point under a random permutation is uniform, so the output that the adversary gets to see is perfectly uniform.

For the authenticity bound, let the queries of $\mathcal{A}$ be $(N^{(i)}, X^{(i)})$, for $1 \le i \le q-1$ and in response it gets $Y^{(i)}$. Finally, $\mathcal{A}$ outputs the forgery $(N^{(q)}, X^{(q)}; Y^{(q)})$. We need to bound the probability that this forgery is successful. We have the following two cases to consider.

Case 1: $N^{(q)} \notin \{N^{(1)}, N^{(2)}, \cdots, N^{(q-1)}\}$. Note, in this case the final output for the tag is an image of the permutation $\pi^{0,N^{(q)}}$ and the adversary has never seen any image of this permutation in its previous queries. Thus irrespective of the choice of $X^{(q)}$ and $Y^{(q)}$, $\mathcal{A}$ succeeds with probability $1/2^n$.

Case 2: $N^{(q)} \in \{N^{(1)}, N^{(2)}, \cdots, N^{(q-1)}\}$. Let us consider that the j^{th} query of $\mathcal{A}$ has the same nonce as its forgery, i.e., $N^{(q)} = N^{(j)}$. It would be enough for us to concentrate on the query $(N^{(j)}, X^{(j)})$ and its response $Y^{(j)}$ to analyze the success probability of $\mathcal{A}$, as the responses to the other queries are unrelated to the task of the adversary of producing a forgery with the nonce $N^{(q)}$. Let $X^{(j)} = X_1^{(j)} \| X_2^{(j)} \| \ldots \| X_m^{(j)}$, and $X^{(q)} = X_1^{(q)} \| X_2^{(q)} \| \cdots \| X_{m'}^{(q)}$. For the j^{th} query, $(N^{(j)}, X^{(j)})$, we have

$$(\mathsf{sum}')^{(j)} = Z_1^{(j)} \oplus Z_2^{(j)} \oplus \cdots Z_m^{(j)},$$

and the tag $Y^{(j)} = \pi^{0,N^{(j)}}\left((\mathsf{sum}')^{(j)}\right)$. Each $Z_i^{(j)} = \left(\pi^{i,N^{(j)}}\right)^{-1}(X_i^{(j)})$. As $\pi^{i,N}$ is an uniform random permutation, hence each $Z_i^{(j)}$ is uniformly distributed. Thus $(\mathsf{sum}')^{(j)}$ is uniformly distributed in $\{0,1\}^n$.

For the forgery, we have

$$(\mathsf{sum}')^{(q)} = Z_1^{(q)} \oplus Z_2^{(q)} \oplus \cdots Z_{m'}^{(q)}.$$

For the forgery to be successful, it is required that $Y^{(q)} = \pi^{0,N^{(j)}}\left((\mathsf{sum}')^{(q)}\right)$. We consider two sub-cases.

- *Case 2A:* $Y^{(j)} = Y^{(q)}$. In this case, the forgery is successful only if $(\mathsf{sum}')^{(j)} = (\mathsf{sum}')^{(q)}$. For the forgery to be valid, it is required that $X^{(j)} \neq X^{(q)}$ which means that they differ in at least one block. We consider two possibilities:
  - If $X^{(q)}$ is not a prefix of $X^{(j)}$, then there exists a $k \le \min(m, m')$ such that $X_k^{(q)} \neq X_k^{(j)}$. Thus, $\left(\pi^{k,N^{(q)}}\right)^{-1}(X_k^{(q)})$ is uniform random. This implies that $(\mathsf{sum}')^{(j)} \oplus (\mathsf{sum}')^{(q)}$ is uniformly distributed in $\{0,1\}^n$.
  - If $X^{(q)}$ is a prefix of $X^{(j)}$, then for every k, where $m' < k \le m$, $\left(\pi^{k,N^{(j)}}\right)^{-1}(X_k^{(j)})$ is uniform random, thus making $(\mathsf{sum}')^{(j)} \oplus (\mathsf{sum}')^{(q)}$ uniformly distributed in $\{0,1\}^n$.

  Thus, for both the above possibilities, $\Pr[(\mathsf{sum}')^{(j)} = (\mathsf{sum}')^{(q)}] = 1/2^n$, which is the success probability of $\mathcal{A}$.
- *Case 2B:* $Y^{(j)} \neq Y^{(q)}$. In this case, for the forgery to be successful $(\mathsf{sum}')^{(j)} \neq (\mathsf{sum}')^{(q)}$. $\mathcal{A}$ already knows the image of $(\mathsf{sum}')^{(j)}$, under the random permutation $\pi^{0,N^{(j)}}$ and with this knowledge it is trying to guess the image of another point under the same (random) permutation. The probability of guessing this correctly is $1/(2^n - 1)$, which in turn is its success probability.

■

Proposition 2: There exists an adversary $\mathcal{A}$ which asks only one query consisting of two n-bit blocks of message such that

$$\mathbf{Adv}^{\text{ae-auth}}_{\mathsf{insOCB}}(\mathcal{A}) = 1.$$

Proof: We construct $\mathcal{A}$. $\mathcal{A}$ asks the query $(N, P \| 0^n)$, where N, P are any n-bit strings. In response to this query $\mathcal{A}$ gets $(C_1 \| C_2; \mathsf{tag})$. Finally $\mathcal{A}$ produces the forgery $(N, C_1; \mathsf{tag})$.

From the description of insOCB, we have

$$\begin{aligned} C_1 &= \pi^{1,N}(P), \\ C_2 &= \pi^{2,N}(0^n), \\ \mathsf{tag} &= \pi^{0,N}(P_1 \oplus 0^n) = \pi^{0,N}(P_1) \end{aligned} \tag{11}$$

Now, when the forgery attempt $(N, C_1; \mathsf{tag})$ is decrypted, the internal variable sum' takes the following value

$$\mathsf{sum}' = \left(\pi^{1,N}\right)^{-1}(C_1) = P_1,$$

and

$$\mathsf{tag}' = \pi^{0,N}(\mathsf{sum}') = \pi^{0,N}(P_1). \tag{12}$$

Thus, from (11) and (12), we see that the forgery is always successful. ■

Thus the construction of insOCB along with the Propositions 1 and 2 implies that (8) is false.

This counterexample for both (6) and (8) gives us some interesting insights:

1) An adversary attacking the authenticity of an AE scheme f has much more information than the adversaries attacking the privacy of f or the authenticity of $\tilde{f}$. This becomes very clear through the analysis for insOCB that we present here. In case of insOCB the privacy adversary always sees random strings, and the adversary attacking the authenticity of $\widetilde{\mathsf{insOCB}}$ only sees some images of the permutation $\pi^{0,N}$ for different values of N. We argue in Proposition 1 that this information does not help it in anyway to produce a forgery. On the other hand, an adversary attacking the authenticity of insOCB gets to see the ciphertexts corresponding to its chosen plaintexts, this additional information helps it to mount a forgery attack on insOCB.
2) It seems unlikely that one can express the authenticity security of f in terms of its privacy security and the authenticity security of $\tilde{f}$.

V. Final Remarks

1) We show attacks on PAE and thus show it to be completely insecure. A variant of PAE called PAE-1 is also proposed in [6], similar attacks also applies for PAE-1. The AEAD constructions PAEAD and PAEAD-1 which are constructed using PAE and PAE-1, respectively, are also insecure.
2) A message authentication code named iPMAC was also proposed in [6]. Our attacks do not affect the security of iPMAC and it seems to be secure.
3) The Proposition 6 in [6] is false. The error in the proof stems from some oversights, and the implicit assumption of a reduction which possibly does not exist. This wrong proposition allowed the author of [6] to prove security of an insecure construction.
4) A recent work [3] proposes new MAC, AE and AEAD constructions which are related to the constructions in [6]. [3] includes rigorous security analysis and implementational studies of the proposed schemes, which seems to be secure.

Acknowledgements

The authors thank Palash Sarkar and the anonymous reviewers for their comments and suggestions. Debrup Chakraborty acknowledges the support from Consejo Nacional de Ciencia y Tecnología (CONACyT), Mexico, through the project 166763.

References

[1] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.

[2] CAESAR. Competition for Authenticated Encryption: Security, Applicability, and Robustness. http://competitions.cr.yp.to/caesar.html.

[3] Debrup Chakraborty and Palash Sarkar. On modes of operations of a block cipher for authentication and authenticated encryption. *IACR Cryptology ePrint Archive*, 2014:627, 2014.

[4] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In Antoine Joux, editor, *FSE*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327. Springer, 2011.

[5] Phillip Rogaway, Mihir Bellare, and John Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Trans. Inf. Syst. Secur.*, 6(3):365–403, 2003.

[6] Palash Sarkar. Pseudo-random functions and parallelizable modes of operations of a block cipher. *IEEE Transactions on Information Theory*, 56(8):4025–4037, 2010.

Debrup Chakraborty received the BE degree in mechanical engineering from Jadavpur University, Kolkata, India, in 1997, and the MTech and PhD degrees in computer science from the Indian Statistical Institute, Kolkata, in 1999 and 2005, respectively. Currently, he is a researcher in the Computer Science Department of Centro de Investigación y de Estudios Avanzados del IPN, Mexico City, Mexico. His current research interests include design and analysis of provably secure symmetric encryption schemes, efficient software/hardware implementations of cryptographic primitives, pattern recognition, and neural networks..

Mridul Nandi received the B.Stat and M.Stat degrees from Indian Statistical Institute, Kolkata, India, in 1999 and 2001, respectively. He also obtained his PhD degree in computer science from the Indian Statistical Institute, Kolkata, in 2005. Since 2011 till now, he is an assistant professor in the Applied Statistics Unit, Indian Statistical Institute, Kolkata. His current research interests include design, security proof and cryptanalysis of symmetric encryption schemes; efficient software/hardware implementations of cryptographic algorithms; functional encryption schemes and signature schemes.