

1 current sec model to transcribe ADEM + AMAC to a more symmetric style

1.1 used primitives

- ADEM: input tag, key and message lead to a cythertext. It should be improbable distinguish the cythertexts of two messages. (adversary may choose two cythertexts and has to guess which one of the two is encrypted)
- AMAC: input tag, key and message lead to a cythertext. It should be improbable to make a forgery (a pair (key, tag, message, cythertext) that verifies without begin generated by calling Omac(key, tag, message) first)

1.2 goal

Each user is provided with two keys, a message and a tag that is bound to the user and does not repeat between users. The message is encrypted using the tag and two keys to generate a cythertext consisting of two parts. First part is Cdem which is the message encrypted with the nonce and the first key while the second part is Cmac that is the mac computed over Cdem, the tag and the second key. Given only one queries to Oenc per user and multiple queries to Odec which always occurs after the Oenc queries, the message should be protected against active adversaries as long as ADEM and AMAC are secure.

1.3 Sec model

the security is purely based on the games for the AMAC and ADEM that are visible below. All variables are elaborated in the paper

Game N-MIOT-UF _{A,N}	Oracle Omac(j, t, m)	Oracle Ovrf(j, m, c)
00 forged $\leftarrow 0$	07 if $C_j \neq \emptyset$: return \perp	13 if $C_j = \emptyset$: return \perp
01 $T \leftarrow \emptyset$	08 if $t \in T$: return \perp	14 if $(m, c) \in C_j$: return \perp
02 for all $j \in [1..N]$:	09 $T \leftarrow T \cup \{t\}; t_j \leftarrow t$	15 if M.vrf(K_j, t_j, m, c):
03 $K_j \xleftarrow{\$} \mathcal{K}$	10 $c \leftarrow \text{M.mac}(K_j, t_j, m)$	16 forged $\leftarrow 1$
04 $C_j \leftarrow \emptyset$	11 $C_j \leftarrow C_j \cup \{(m, c)\}$	17 return true
05 run A	12 return c	18 return false
06 return forged		

Game N-MIOT-IND _{A,N} ^b	Oracle Oenc(j, t, m_0, m_1)	Oracle Odec(j, c)
00 $T \leftarrow \emptyset$	06 if $C_j \neq \emptyset$: return \perp	12 if $C_j = \emptyset$: return \perp
01 for all $j \in [1..N]$:	07 if $t \in T$: return \perp	13 if $c \in C_j$: return \perp
02 $K_j \xleftarrow{\$} \mathcal{K}$	08 $T \leftarrow T \cup \{t\}; t_j \leftarrow t$	14 $m \leftarrow \text{A.dec}(K_j, t_j, c)$
03 $C_j \leftarrow \emptyset$	09 $c \leftarrow \text{A.enc}(K_j, t_j, m_b)$	15 return m
04 $b' \xleftarrow{\$} \mathcal{A}$	10 $C_j \leftarrow C_j \cup \{c\}$	
05 return b'	11 return c	

with

Proc A.enc'(K, t, m)	Proc A.dec'(K, t, c)
00 ($K_{\text{dem}}, K_{\text{mac}} \leftarrow K$	05 ($K_{\text{dem}}, K_{\text{mac}} \leftarrow K$
01 $c_{\text{dem}} \leftarrow \text{A.enc}(K_{\text{dem}}, t, m)$	06 ($c_{\text{dem}}, c_{\text{mac}} \leftarrow c$
02 $c_{\text{mac}} \leftarrow \text{M.mac}(K_{\text{mac}}, t, c_{\text{dem}})$	07 if M.vrf($K_{\text{mac}}, t, c_{\text{dem}}, c_{\text{mac}}$):
03 $c \leftarrow (c_{\text{dem}}, c_{\text{mac}})$	08 $m \leftarrow \text{A.dec}(K_{\text{dem}}, t, c_{\text{dem}})$
04 return c	09 return m
	10 return \perp

2 needed sec model to transcribe ADEM + AMAC to a more symmetric style

for now we only look at the nonce based options as the pkc paper does that too.

2.1 used primitives

- ADEM: input nonce, key en message lead to a cythertext which should be improbable to distinguish from RO (adversary has to guess if he is talking to RO or ADEM)
- AMAC: input nonce, key en message lead to a tag that should be improbable to distinguish from random oracle (adversary has to guess if he is talking to RO or AMAC)

2.2 goal

Each user is provided with two keys, a message and a lock that does not repeat between users. The message is encrypted using the lock and two keys. Given only one queries to Oenc per user and multiple queries to Odec which always occurs after the Oenc queries, the message should be protected against active adversaries as long as ADEM and AMAC are secure.

2.3 Sec model

We define the following sec games for the AMAC, the ADEM and the ADEM+AMAC (names will prob be improved later):

Game $\text{AMAC}_{A,N}^M$	Oracle $\text{Omac}(j,l,m)$
0 : $L \leftarrow \emptyset$	5 : if $T_j \neq \emptyset$: return \perp
1 : for $j \in [1..N]$:	6 : if $l \in L$: return \perp
2 : $K_j \xleftarrow{\$} K$	7 : $L \leftarrow L \cup \{l\}$
3 : $b' \leftarrow A$	8 : $l_j = l$
4 : return b'	9 : $t \leftarrow M.\text{mac}(K_j, l_j, m)$
	10 : return t

Figure 1: AMAC game, M is either the MAC or Random Oracle

Game $\text{ADEM}_{A,N}^E$	Oracle $\text{Omac}(j,l,m)$
0 : $L \leftarrow \emptyset$	5 : if $T_j \neq \emptyset$: return \perp
1 : for $j \in [1..N]$:	6 : if $l \in L$: return \perp
2 : $K_j \xleftarrow{\$} K$	7 : $L \leftarrow L \cup \{l\}$
3 : $b' \leftarrow A$	8 : $l_j = l$
4 : return b'	9 : $c \leftarrow E.\text{enc}(K_j, l_j, m)$
	10 : return c

Figure 2: ADEM game, E is either the DEM or Random Oracle

Game $AE_{A,N}^{AE}$	Oracle $O_{enc}(j,l,m)$	Oracle $O_{dec}(j,m)$
0 : $L \leftarrow \emptyset$	6 : if $T_j \neq \emptyset$: return \perp	13 : if $c_j \neq \emptyset$: return \perp
1 : for $j \in [1..N]$:	7 : if $l \in L$: return \perp	14 : if $c \in C_j$: return \perp
2 : $K_j \xleftarrow{\$} K$	8 : $L \leftarrow L \cup \{l\}$	15 : $m \leftarrow EA.dec(K_j, L_j, c)$
3 : $C_j \leftarrow \emptyset$	9 : $l_j = l$	16 : return m
4 : $b' \leftarrow A$	10 : $c \leftarrow EA.enc(K_j, l_j, m)$	
5 : return b'	11 : $C_j \leftarrow C_j \cup c$	
	12 : return t	

Figure 3: AE game, where AE is either the AE scheme having access to the MAC and DEM or RO

We should consider what the RO calls should do, there are several cases to consider:

- RO replacing M: $RO.mac(k,l,m)$ calls $t = M.mac(k,l,m)$ then outputs \perp if t is \perp or $|t|$ random bits otherwise.
- RO replacing E: $RO.enc(k,l,m)$ calls $c = E.enc(k,l,m)$ then outputs \perp if c is \perp or $|c|$ random bits otherwise.
- RO replacing AE: $RO.enc(k,l,m)$ calls $c = AE.enc(k,l,m)$ then outputs \perp if c is \perp or $|c|$ random bits otherwise. $RO.dec(k,l,c)$ always returns \perp .

The AE schemes should be constructed from the DEM and the MAC. Following General Composition reconsidered, three ways to construct this EA are of interest, namely the ones following from the N1, N2 and N3 scheme. One thing to keep in mind with this that these schemes would originally use associated data. For now we can discard this but it is not proven that the same security results would also follow from this case without associated data. Down here the initial schemes can be found, followed by the $AE.enc$ and $AE.dec$ calls that can we construct following these schemes.

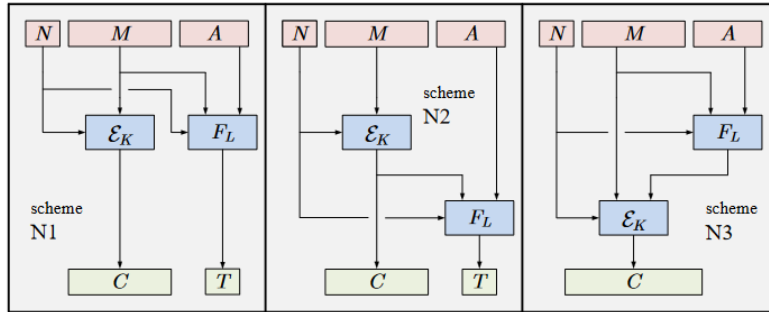


Figure 4: Original N schemes from Generic Composition reconsidered

EA.enc(k,l,m)	EA.dec(k,l,c)
0 : $(k1, k2) \leftarrow k$	5 : $(k1, k2) \leftarrow k$
1 : $c' = E.enc(k1, l, m)$	6 : $(c', t) \leftarrow c$
2 : $t = M.mac(k2, l, m)$	7 : $m = E.dec(k1, l, c')$
3 : $c = (c', t)$	8 : $t' = M.mac(k2, l, m)$
4 : return c	9 : if $t = t'$: return m
	10 : else : return \perp

Figure 5: Calls based on N1

EA.enc(k,l,m)	EA.dec(k,l,c)
0 : $(k1, k2) \leftarrow k$	5 : $(k1, k2) \leftarrow k$
1 : $c' = E.enc(k1, l, m)$	6 : $(c', t) \leftarrow c$
2 : $t = M.mac(k2, l, c')$	7 : $m = E.dec(k1, l, c')$
3 : $c = (c', t)$	8 : $t' = M.mac(k2, l, c')$
4 : return c	9 : if $t = t'$: return m
	10 : else : return \perp

Figure 6: Calls based on N2

EA.enc(k,l,m)	EA.dec(k,l,c)
0 : $(k1, k2) \leftarrow k$	5 : $(k1, k2) \leftarrow k$
1 : $t = M.mac(k2, l, m)$	6 : $m' = E.dec(k1, l, c)$
2 : $m' = m t$	7 : $(m, t) \leftarrow m'$
3 : $c = E.enc(k1, l, m')$	8 : $t' = M.mac(k2, l, m)$
4 : return c	9 : if $t = t'$: return m
	10 : else : return \perp

Figure 7: Calls based on N3

3 burning questions

- Q1: Is het oke om de games gebazeerd te hebben op RO
Q2:(klopt het dat RO stickt sterker is dan left-right)
- in GCrec, waarom zijn er bij de n-schemes geen N4 en N5 schemas?
A: die voegen niets toe Q2: klopt dat?

4 current todo's

- in meer detail opschrijven wat de variables zijn van de sec games en wat de aannamens zijn
- crypto.bib kijken
- mayhabs beginnen met schrijven in main

5 main idea

The PKC paper ends with a ADEM + AMAC construction as "solution". The original paper from ENC -i MAC has been revised, so this should prob be revised as well. In general its nice to write down thing in a more "sym crypto" style as we use symmetric primitives. It would probably also be nice to revise it more in general and see what other ways there are to reach the end-goal expected in the PKC paper.