sp-AELM: Sponge based Authenticated Encryption Scheme for Memory Constrained Devices

Megha Agrawal, Donghoon Chang, Somitra Sanadhya

IIIT Delhi

29 Sept 2015
DIAC 2015, Singapore

Presentation Outline

Online Authenticated Encryption

- Authenticated encryption scheme that supports online encryption.
- Online encryption:
  - Doesn't need to know the whole message in advance.
  - C_i can be calculated without knowledge of M_j for any $j > i$.
- Example: OCB, AEGIS, APE etc.
- When we say Online AE, we consider online encryption only.
- Suitable for memory restricted environments.

What about decryption and verification for low memory devices?

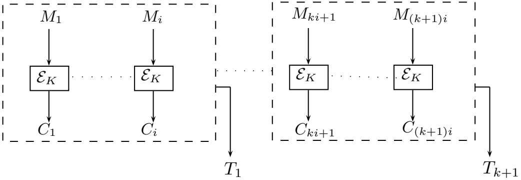
CAESAR solutions

Some of the CAESAR candidate address this issue by using one of the two solutions:

- Intermediate Tag
- Releasing Unverified Plaintext

Intermediate Tag

- A long plaintext is split into separate packets, each of which is separately authenticated (and encrypted), a long forgery need not be buffered before it is rejected.



Disadvantage

- Its not safe if forgery will be at the end.
- Enough buffer space needed for storing multiple tags.

Release Unverified Plaintext(RUP)

- In ASIACRYPT 2014, Andreeval et.al introduced the first formalization of the releasing unverified plaintext (RUP) setting.
- Their scenario assumes that the attacker can see the unverified plaintext, or any information relating to it, before verification is complete.
- They redefine the security notion in RUP settings:
  - For integrity, they propose INT-RUP(integrity under releasing unverified plaintext)
  - For privacy both IND-CPA and PA(plaintext awareness).

Disadvantage

- Adversary may get additional information.
- User of the device may not want to release unverified plaintext.
- Requires an additional security analysis.

Example: if we release unverified plaintext in OCB mode, then it is not secure.

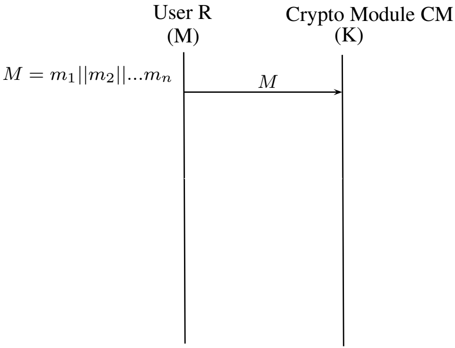
Observations

- Both of these solutions have some trade off between security and efficiency.
- We require some solution that is efficient as well as doesn't compromise with the security.

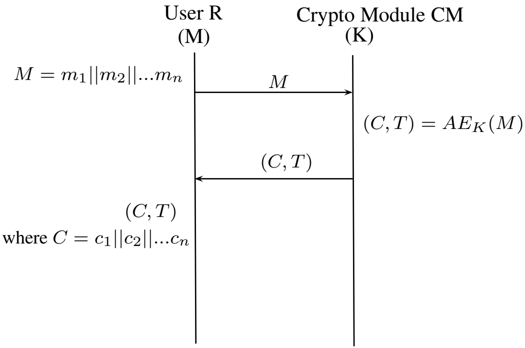
Decrypt-Then-Mask

- Introduced by Fouque et. al in SAC 2003. [2]
- Supports low memory verification
- Main idea is to mask the decrypted text blocks by XORing with pseudorandom sequence of bits and outputting seed used to generate the pseudorandom sequence, if tag is valid.

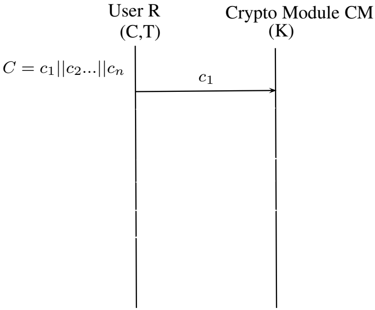
Encryption



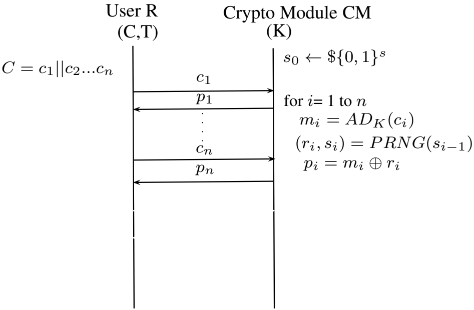
Encryption


User R
(M)
Crypto Module CM
(K)
$M = m_1||m_2||...m_n$
M
$(C, T) = AE_K(M)$
(C, T)
(C, T)
where $C = c_1||c_2||...c_n$

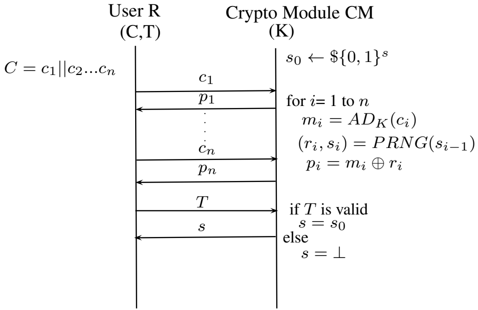
Decryption



Decryption

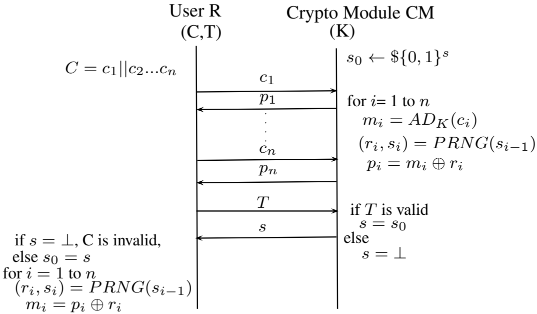

User R
(C,T)
Crypto Module CM
(K)
$C = c_1||c_2...c_n$
$s_0 \leftarrow \$\{0,1\}^s$
c_1
p_1
for i= 1 to n
$m_i = AD_K(c_i)$
$(r_i, s_i) = PRNG(s_{i-1})$
$p_i = m_i \oplus r_i$
c_n
p_n

Decryption



User R
(C,T)
Crypto Module CM
(K)
$C = c_1||c_2...c_n$
$s_0 \leftarrow \$\{0,1\}^s$
c_1
p_1
c_n
p_n
for i= 1 to n
$m_i = AD_K(c_i)$
$(r_i, s_i) = PRNG(s_{i-1})$
$p_i = m_i \oplus r_i$
T
s
if T is valid
$s = s_0$
else
$s = \perp$

Decryption


User R
(C,T)
Crypto Module CM
(K)
$s_0 \leftarrow \$\{0,1\}^s$
$C = c_1||c_2...c_n$
c_1
p_1
for i= 1 to n
$m_i = AD_K(c_i)$
$(r_i, s_i) = PRNG(s_{i-1})$
$p_i = m_i \oplus r_i$
c_n
p_n
T
if T is valid
$s = s_0$
else
$s = \perp$
s
if $s = \perp$, C is invalid,
else $s_0 = s$
for $i = 1$ to n
$(r_i, s_i) = PRNG(s_{i-1})$
$m_i = p_i \oplus r_i$

Drawbacks

- Requires two additional passes due to the usage of pseudorandom number generator.
- Expensive for long messages.
- Communication overhead is more during decryption.

Presentation Outline

Motivation

- All existing solutions to support decryption and verification for low memory devices have some drawbacks.
- No efficient solution exists till now.

Presentation Outline

Research Contribution

- Proposed a new generalized technique that overcomes existing problems.
- We explained our technique through 3 example constructions
- Provide its security proof for Privacy and Authenticity using code based game playing framework.
- Analyze sponge based CAESAR submissions using our proposed technique to determine their suitability for this newly defined scenario.

Generalized Construction

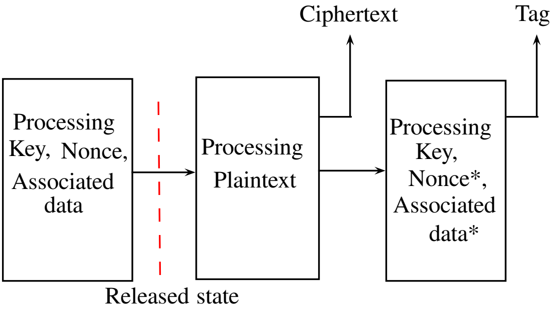


Figure: Block diagram for generalized construction

*processing of nonce and associated data are optional at the end.

How does it work?

- During decryption it just decrypts the ciphertext and doesnt store any block of decrypted text except one intermediate state(shown using red line).
- If at the end tag gets verified then it release the stored intermediate state to user.
- Using this intermediate state, user can compute plaintext at their side.
- Since at the end we are processing key again, so user can not do any forgery.

Examples

Eample 1: sp-AELM

- Input: (K, A, M)
  - K: Key, A: Associated Data, M: Message
  - $M = m_0 \| m_1 \| .... \| m_{n-1}$
- Output: (N, C, T)
  - N: Nonce, C: Ciphertext, T: Tag
  - $C = c_0 \| c_1 \| .... \| c_{n-1}$

sp-AELM construction

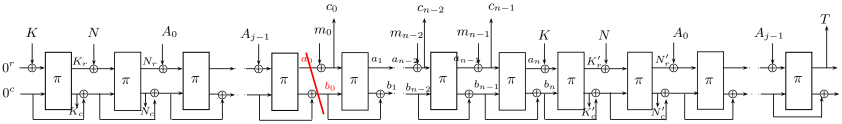
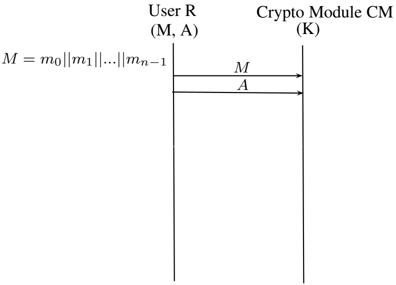
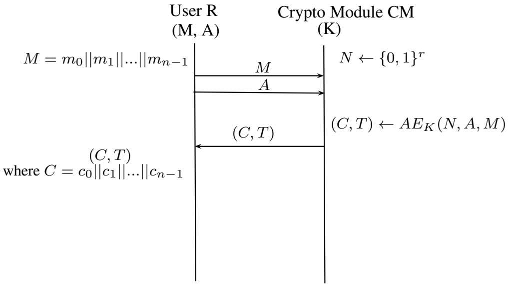


Figure: sp-AELM

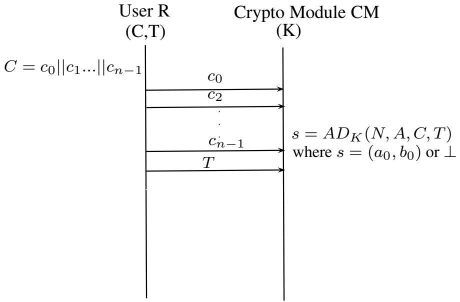
Encryption



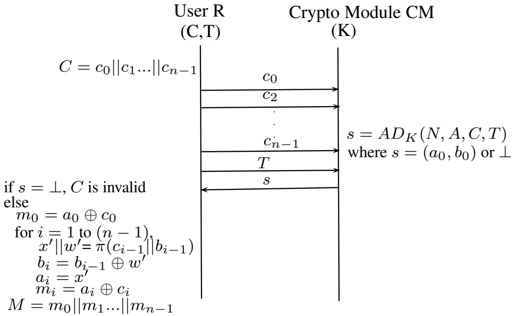
Encryption


User R
(M, A)
Crypto Module CM
(K)
$M = m_0||m_1||...||m_{n-1}$
$N \leftarrow \{0,1\}^r$
M
A
$(C,T) \leftarrow AE_K(N,A,M)$
(C,T)
(C,T)
where $C = c_0||c_1||...||c_{n-1}$

Decryption


User R
(C,T)
Crypto Module CM
(K)
$C = c_0||c_1...||c_{n-1}$
c_0
c_2
c_{n-1}
T
$s = AD_K(N, A, C, T)$
where $s = (a_0, b_0)$ or $\perp$

Decryption


User R
(C,T)
Crypto Module CM
(K)
$C = c_0||c_1...||c_{n-1}$
c_0
c_2
c_{n-1}
T
s
$s = AD_K(N, A, C, T)$
where $s = (a_0, b_0)$ or $\perp$
if $s = \perp$, C is invalid
else
$m_0 = a_0 \oplus c_0$
for $i = 1$ to $(n-1)$,
$x'||w' = \pi(c_{i-1}||b_{i-1})$
$b_i = b_{i-1} \oplus w'$
$a_i = x'$
$m_i = a_i \oplus c_i$
$M = m_0||m_1...||m_{n-1}$

Example 2 & 3

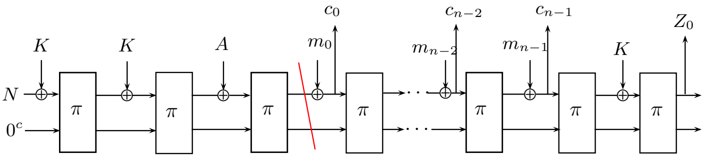


Figure: Variant 1

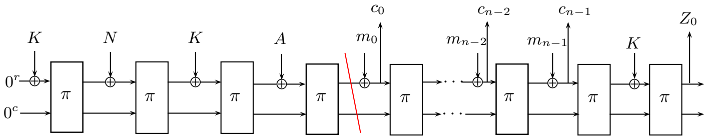


Figure: Variant 2

Features

- Doesn't need to store decrypted text blocks.
- One pass for encryption.
- Two pass for decryption and tag verification.
- Instead of returning all plaintext values, it just give one intermediate state to the user.
    - Plaintext can be calculated at user end.

Presentation Outline

Analysis of Sponge based submissions

- There were 9 Sponge based submissions in CAESAR for first round.
- We analyze those 9 submissions using the same technique as in sp-AELM.
- Out of nine, only 2 namely ASCON and PRIMATEs GIBBON securely satisfied the scenario and rest were not secure.

continued..

Sponge based AE Schemes submitted in CAESAR	**Support for limited memory devices**
Artemia, ICEPOLE, Ketje, Keyak, NORX, PRIMATEs(APE, HANUMAN), STRIBOB, Π-Cipher	No
Ascon, PRIMATEs (GIBBON)	Yes

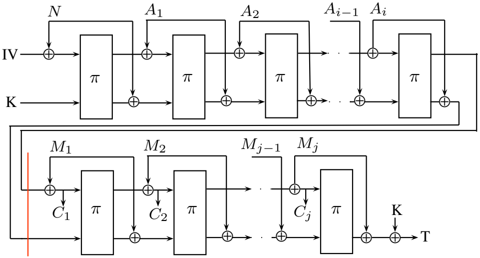


Figure: JHAE mode used in ARTEMIA

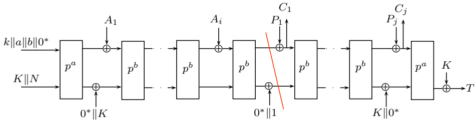


Figure: ASCON

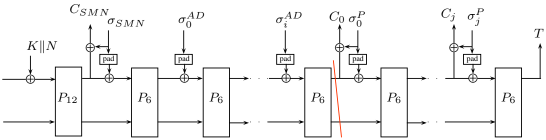


Figure: ICEPOLE

Presentation Outline

Conclusion

- We present the new generalised technique to support decryption and verification for low memory devices.
- We present 3 Sponge based construction using this technique.
- Analyse all sponge based submissions in CAESAR.

Future Work

- We are now trying to apply this proposed technique to Block cipher based AE schemes.

Thank You for your Attention.

References

Yevgeniy Dodis.
Concealment and Its Applications to Authenticated Encryption.
In Alexander W. Dent and Yuliang Zheng, editors, *Practical Signcryption*, Information Security and Cryptography, pages 149–173. Springer, 2010.

Pierre-Alain Fouque, Antoine Joux, Gwenaëlle Martinet, and Frédéric Valette.
Authenticated On-Line Encryption.
In Mitsuru Matsui and Robert J. Zuccherato, editors, *Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 145–159. Springer, 2003.

Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda.
How to securely release unverified plaintext in authenticated encryption.
In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 105–125. Springer, 2014.

Mihir Bellare and Phillip Rogaway.
Code-Based Game-Playing Proofs and the Security of Triple Encryption.
IACR Cryptology ePrint Archive, 2004:331, 2004.