How To Securely Release Unverified Plaintext in Authenticated Encryption

23.08.2014

Elena Andreeva Andrey Bogdanov Atul Luykx
Bart Mennink Nicky Mouha Kan Yasuda

COPA Proposed Second-Round Tweak

Given Nandi's attack on XLS, we intend to tweak COPA by removing XLS and extending the use of tag-splitting from short messages to arbitrary length messages. Formal specification to follow later.

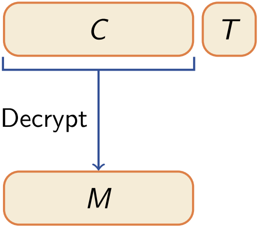
Authenticated Decryption

Decrypt-then-Verify

Verify-then-Decrypt

Authenticated Decryption

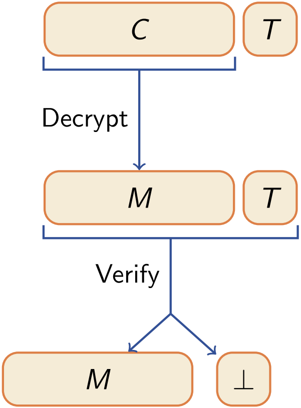
Decrypt-then-Verify



Verify-then-Decrypt

Authenticated Decryption

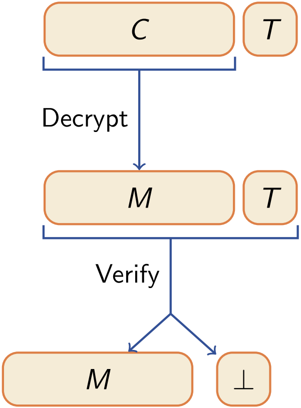
Decrypt-then-Verify



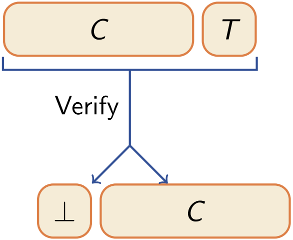
Verify-then-Decrypt

Authenticated Decryption

Decrypt-then-Verify

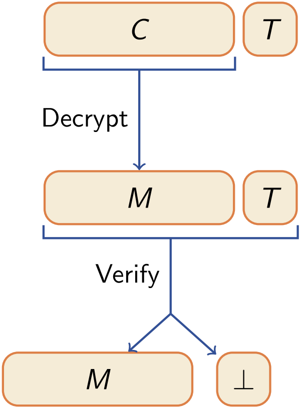


Verify-then-Decrypt

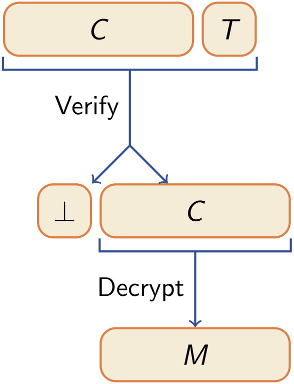


Authenticated Decryption

Decrypt-then-Verify



Verify-then-Decrypt

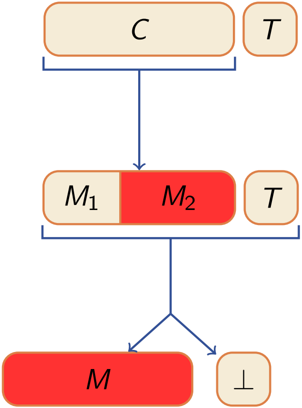


Motivation

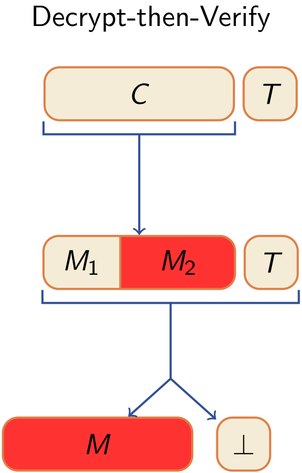
Releasing Unverified Plaintext
3 Scenarios

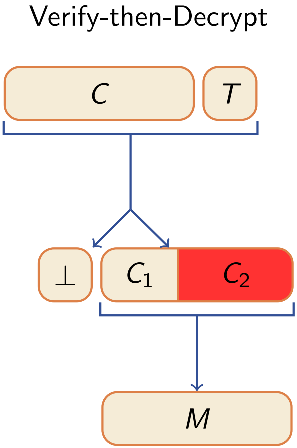
Motivation: Insecure Memory

Decrypt-then-Verify



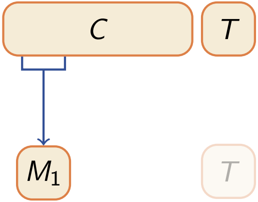
Motivation: Insecure Memory





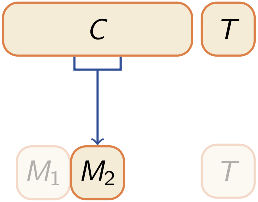
Motivation: Small Buffer

Decrypt-then-Verify



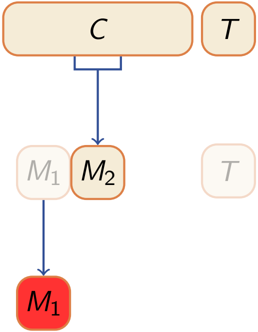
Motivation: Small Buffer

Decrypt-then-Verify



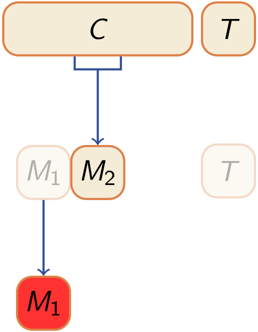
Motivation: Small Buffer

Decrypt-then-Verify

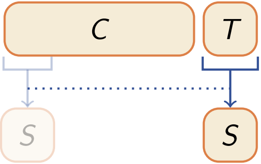


Motivation: Small Buffer

Decrypt-then-Verify

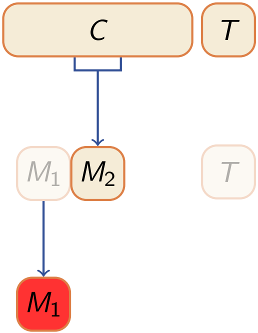


Verify-then-Decrypt

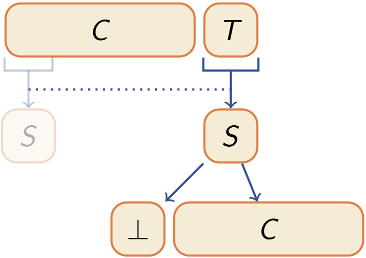


Motivation: Small Buffer

Decrypt-then-Verify

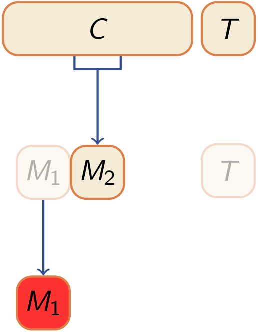


Verify-then-Decrypt

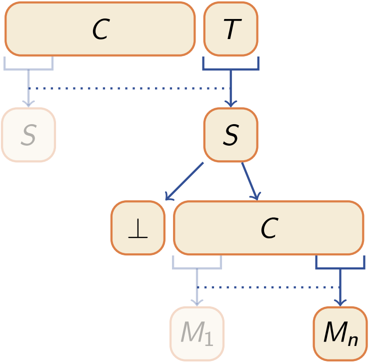


Motivation: Small Buffer

Decrypt-then-Verify

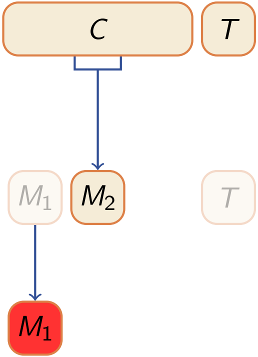


Verify-then-Decrypt

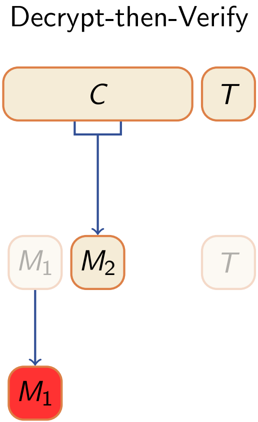


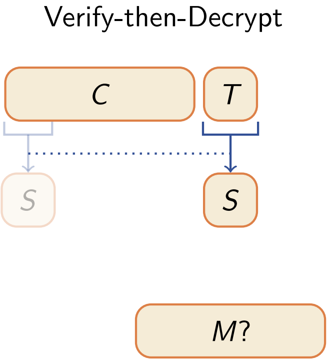
Motivation: Real-time Output

Decrypt-then-Verify



Motivation: Real-time Output



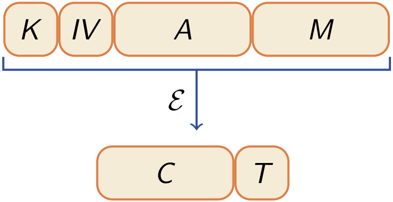


Outline

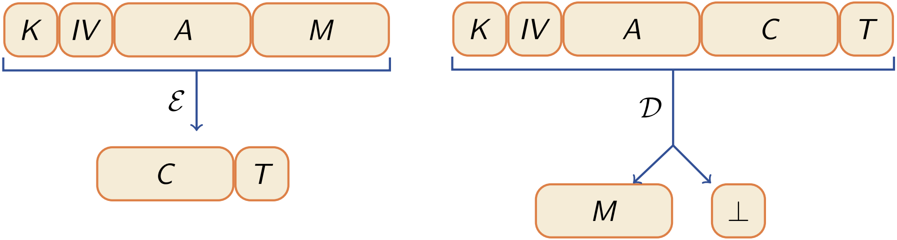
Formalization of RUP

Integrity and Confidentiality

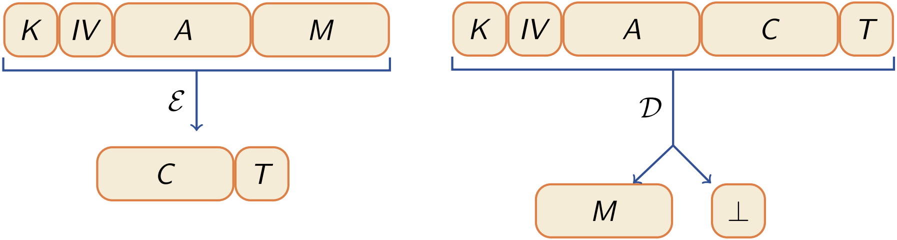
Conventional AE Syntax



Conventional AE Syntax



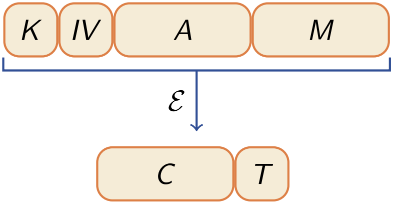
Conventional AE Syntax



IV:

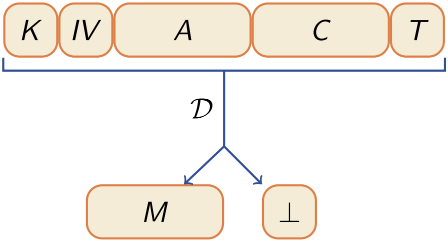
1. random
2. nonce
3. arbitrary

Conventional AE Syntax



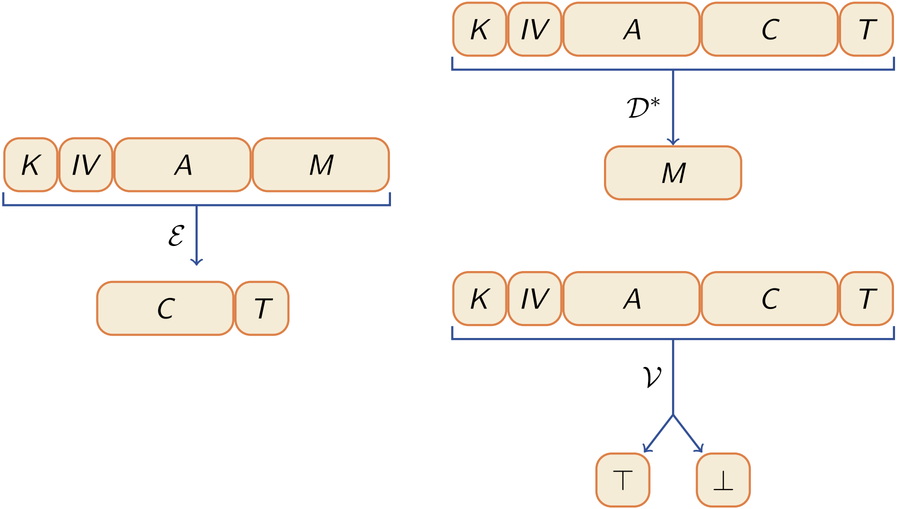
IV:

1. random
2. nonce
3. arbitrary



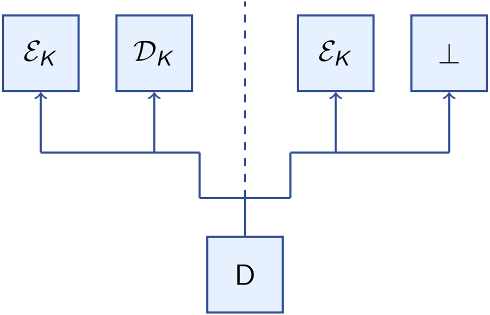
IV always arbitrary

Split AE Syntax



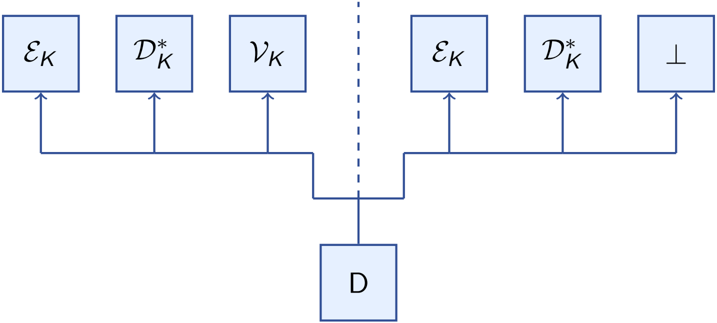
Conventional Integrity

INT-CTXT



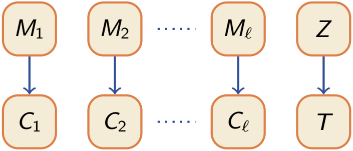
RUP Integrity

INT-RUP



OCB Attack

1. $\mathcal{E}_K$-query, $Z := \bigoplus_{i=1}^{\ell} M_i$



OCB Attack

1. $\mathcal{E}_K$-query, $Z := \bigoplus_{i=1}^{\ell} M_i$

$$\begin{tikzcd}
M_1 \arrow[d] & M_2 \arrow[d] & \cdots & M_\ell \arrow[d] & Z \arrow[d] \\
C_1 & C_2 & \cdots & C_\ell & T
\end{tikzcd}$$

2. Two $\mathcal{D}_K^*$-queries, $j = 0, 1$:

$$\begin{tikzcd}
C_1^j \arrow[d] & C_2^j \arrow[d] & \cdots & C_\ell^j \arrow[d] \\
M_1^j & M_2^j & \cdots & M_\ell^j
\end{tikzcd}$$

$C_i \neq C_i^j, C_i^1 \neq C_i^0$ for all i, j

OCB Attack

1. $\mathcal{E}_K$-query, $Z := \bigoplus_{i=1}^{\ell} M_i$

$$\begin{tikzcd} M_1 \arrow[d] & M_2 \arrow[d] & \cdots & M_\ell \arrow[d] & Z \arrow[d] \\ C_1 & C_2 & \cdots & C_\ell & T \end{tikzcd}$$

2. Two $\mathcal{D}_K^*$-queries, $j = 0, 1$:

$$\begin{tikzcd} C_1^j \arrow[d] & C_2^j \arrow[d] & \cdots & C_\ell^j \arrow[d] \\ M_1^j & M_2^j & \cdots & M_\ell^j \end{tikzcd}$$

$C_i \neq C_i^j, C_i^1 \neq C_i^0$ for all i, j

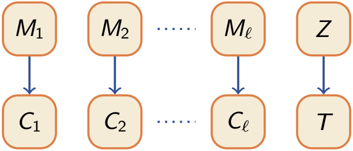
3. Solve system of equations

$$\begin{pmatrix} z_1' \\ z_2' \\ \vdots \\ z_n' \end{pmatrix} = \begin{pmatrix} M_1^0 \oplus M_1^1 & \cdots & M_\ell^0 \oplus M_\ell^1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_\ell \end{pmatrix}$$

where $Z' = \bigoplus_{i=1}^{\ell} M_i^1 \oplus Z$

OCB Attack

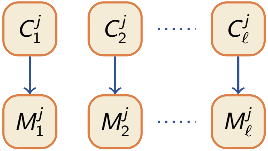
1. $\mathcal{E}_K$-query, $Z := \bigoplus_{i=1}^{\ell} M_i$



M_1 M_2 …… M_ℓ Z

C_1 C_2 …… C_ℓ T

2. Two $\mathcal{D}_K^*$-queries, $j = 0, 1$:



C_1^j C_2^j …… C_ℓ^j

M_1^j M_2^j …… M_ℓ^j

$C_i \neq C_i^j, C_i^1 \neq C_i^0$ for all i, j

3. Solve system of equations

$$\begin{pmatrix} z_1' \\ z_2' \\ \vdots \\ z_n' \end{pmatrix} = \begin{pmatrix} M_1^0 \oplus M_1^1 & \cdots & M_\ell^0 \oplus M_\ell^1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_\ell \end{pmatrix}$$

where $Z' = \bigoplus_{i=1}^{\ell} M_i^1 \oplus Z$

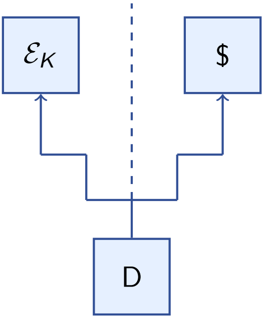
4. Submit forgery:

$$C' = C_1^{x_1} C_2^{x_2} \cdots C_\ell^{x_\ell}$$

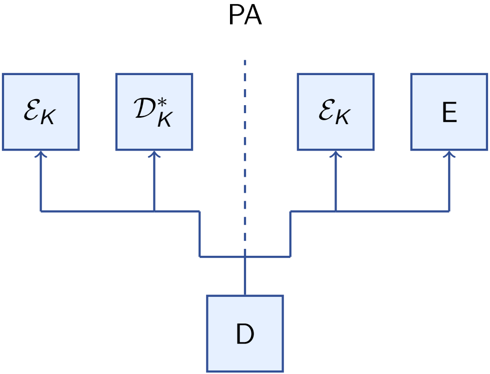
Probability of success at least
$1 - 2^{n-\ell}$

Confidentiality

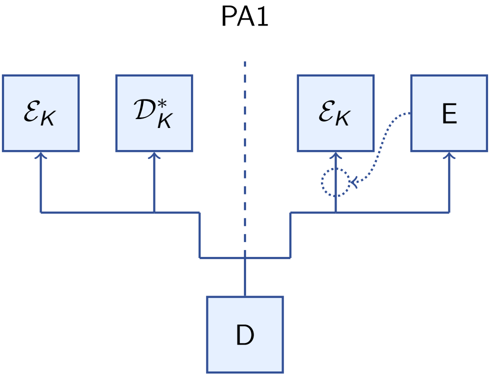
IND-CPA



Plaintext Awareness

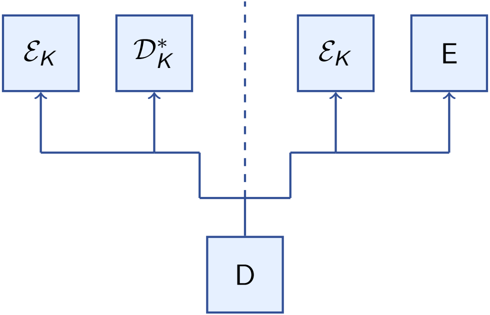


Plaintext Awareness

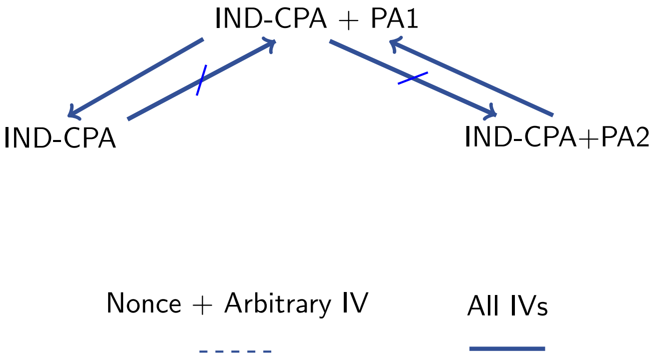


Plaintext Awareness

PA2



Relations



IND-CPA + PA1

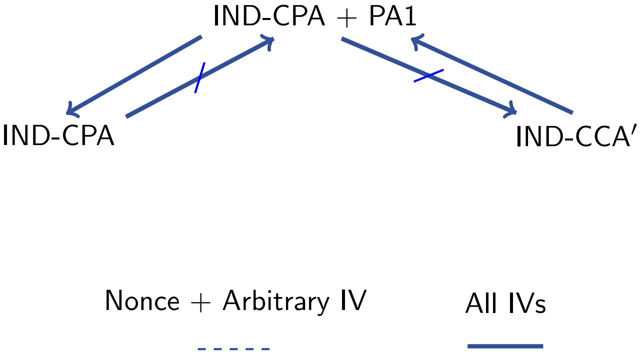
IND-CPA

IND-CPA+PA2

Nonce + Arbitrary IV

All IVs

Relations



IND-CPA + PA1

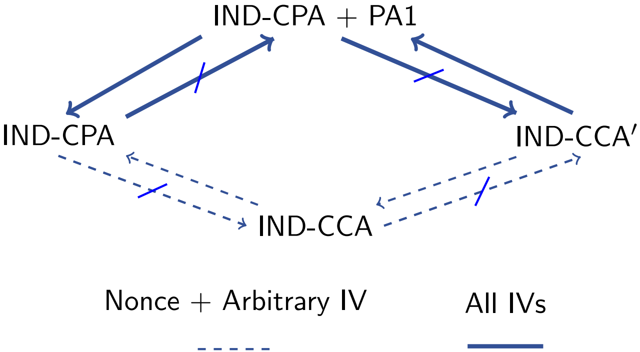
IND-CPA

IND-CCA′

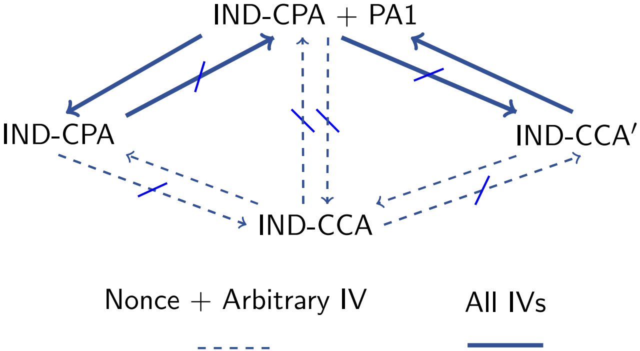
Nonce + Arbitrary IV

All IVs

Relations



Relations



PA Classification of Schemes

IV type	Online	Scheme	PA1	PA2	Remark
random	✓	CTR, CBC	✓	✗	
nonce	✓	OCB	✗	✗	
	✓	GCM, SpongeWrap	✗	✗	
	✗	CCM	✗	✗	not online
arbitrary	✓	COPA	✗	✗	privacy up to prefix
	✓	McOE-G	✗	✗	"
	✓	APE	✓	✗	", backwards decryption
	✗	SIV, BTM, HBS	✓	✗	privacy up to repetition
	✗	Encode-then-Encipher	✓	✓	", VIL SPRP, padding

Conclusions

Formalization

1. Integrity: INT-RUP
2. Confidentiality: IND-CPA + PA1/2

Analysis

1. OCB, COPA INT-RUP attack
2. Relations among PA notions
3. Classification via PA1/2

More can be found in `http://eprint.iacr.org/2014/144`