How to Abuse and Fix Authenticated Encryption Without Key Commitment

Ange Albertini[1], Thai Duong[1], Shay Gueron[2,3], **Stefan Kölbl[1]**, Atul Luykx[1], Sophie Schmieg[1]

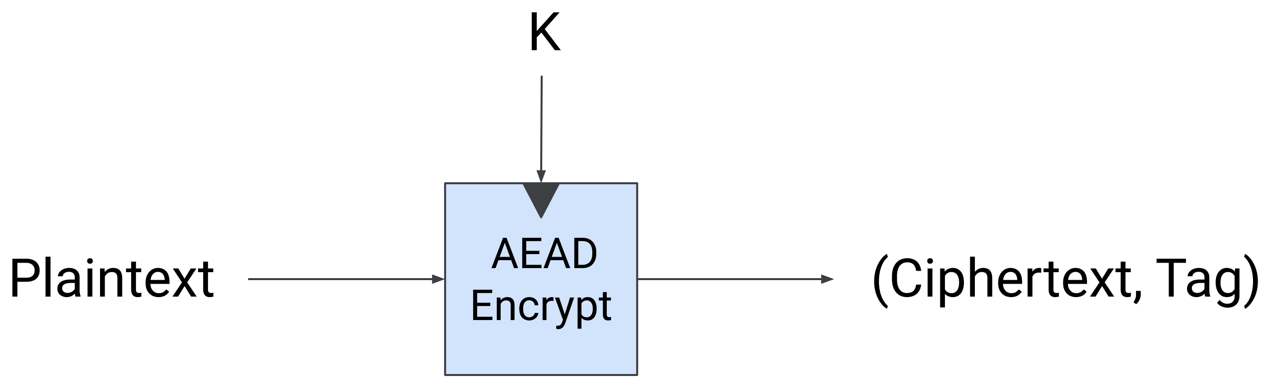
[1]Security Engineering Research, Google

[2]University of Haifa

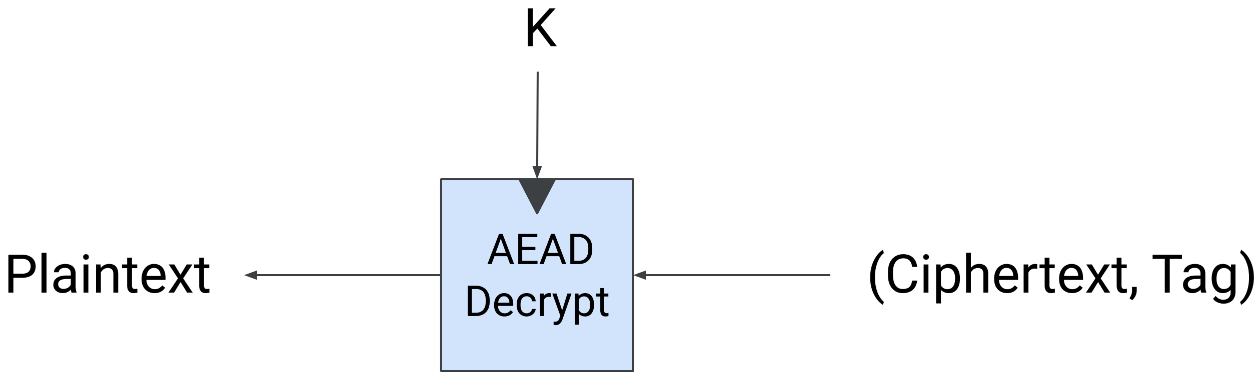
[3]Amazon

What is Key Commitment?

What is Key Commitment?

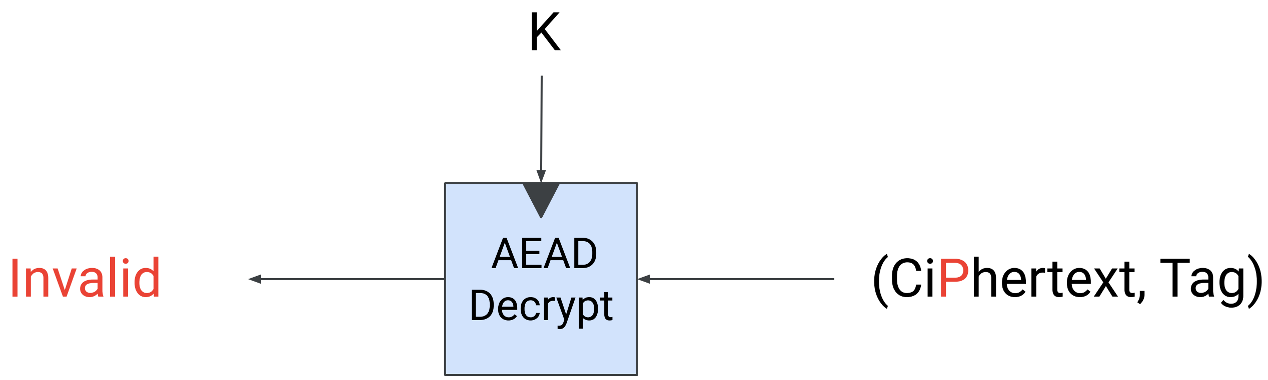


What is Key Commitment?

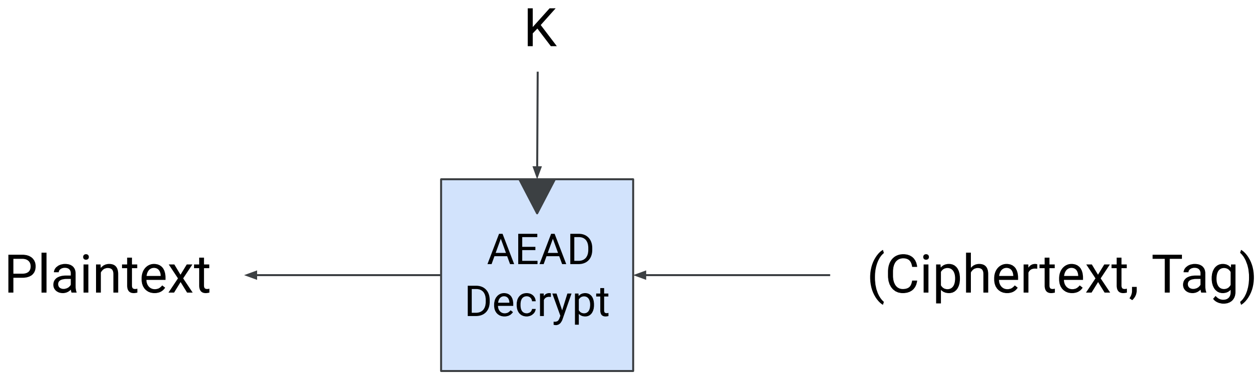


Google

What is Key Commitment?

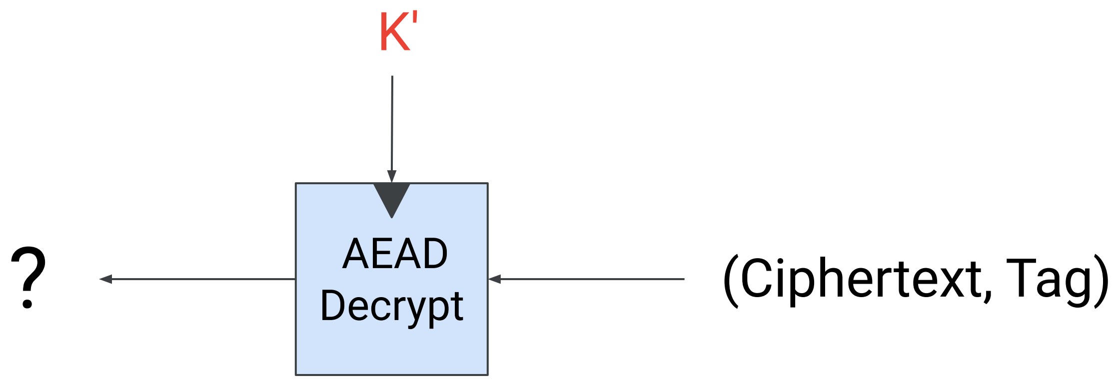


What is Key Commitment?



Google

What is Key Commitment?



Contributions

- Explore vulnerable settings and products.
- Study practical ways to exploit lack of key commitment.
- Provide simple and efficient ways to add key commitment.

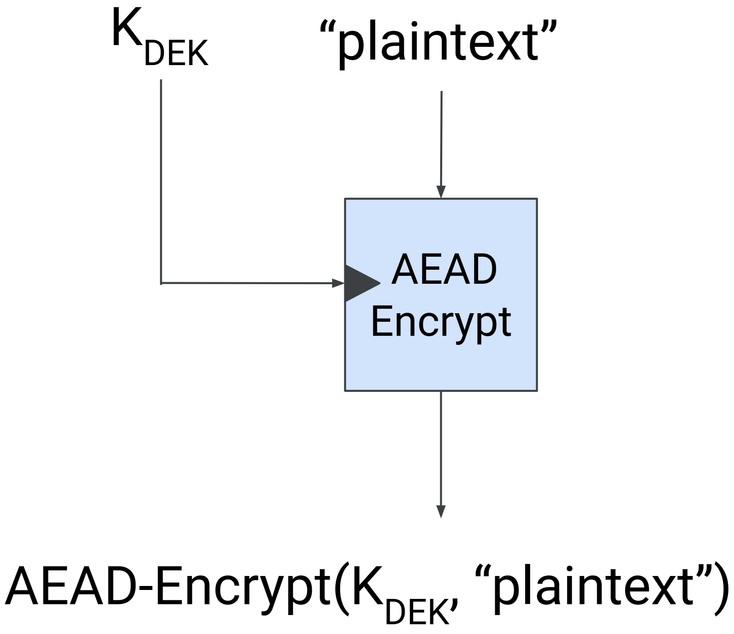
Case Study: Envelope Encryption

Case Study: Envelope Encryption

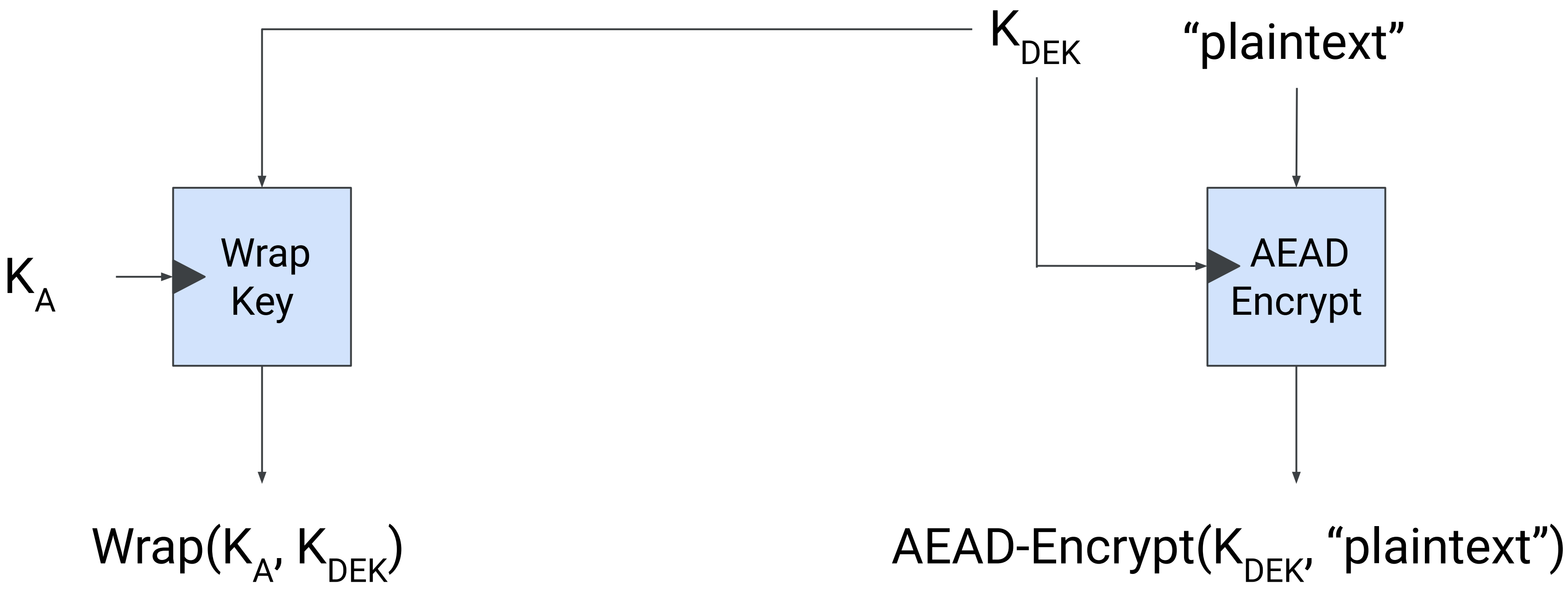
Envelope Encryption

- All major cloud service providers use envelope encryption.
- Encrypt data with symmetric key (DEK), and wrap DEK under multiple symmetric or asymmetric recipient keys (KEK).

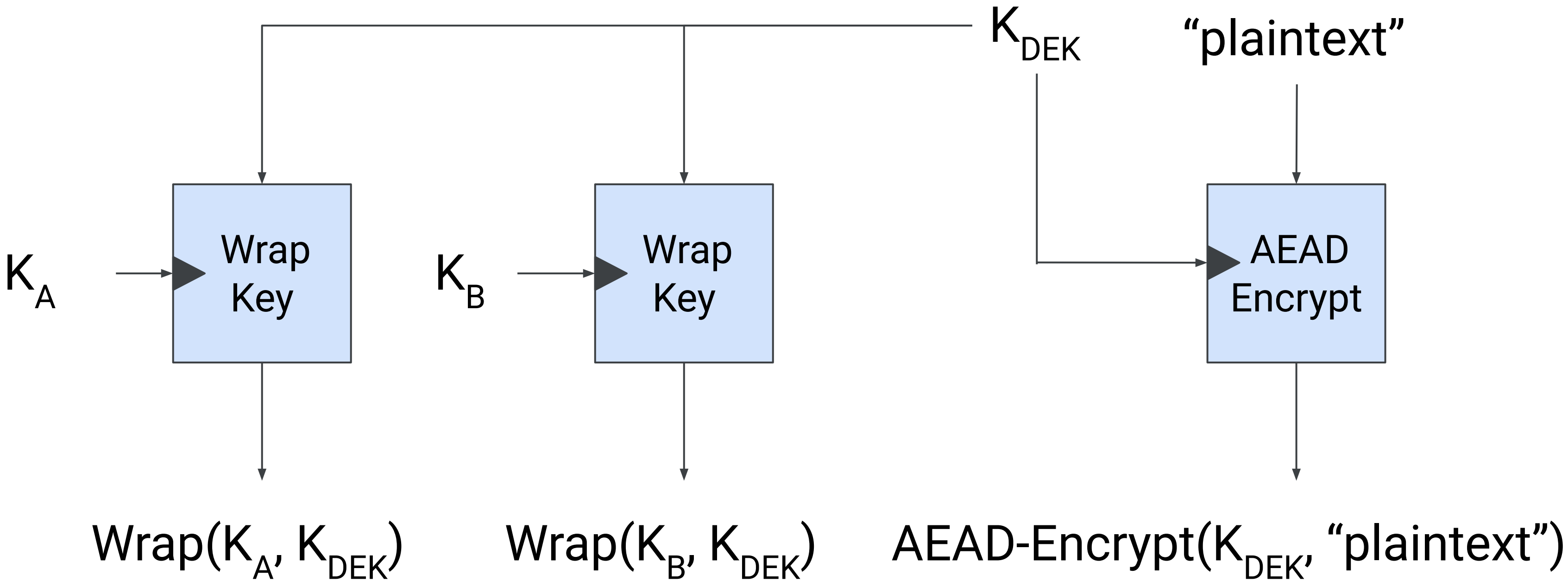
Case Study: Envelope Encryption



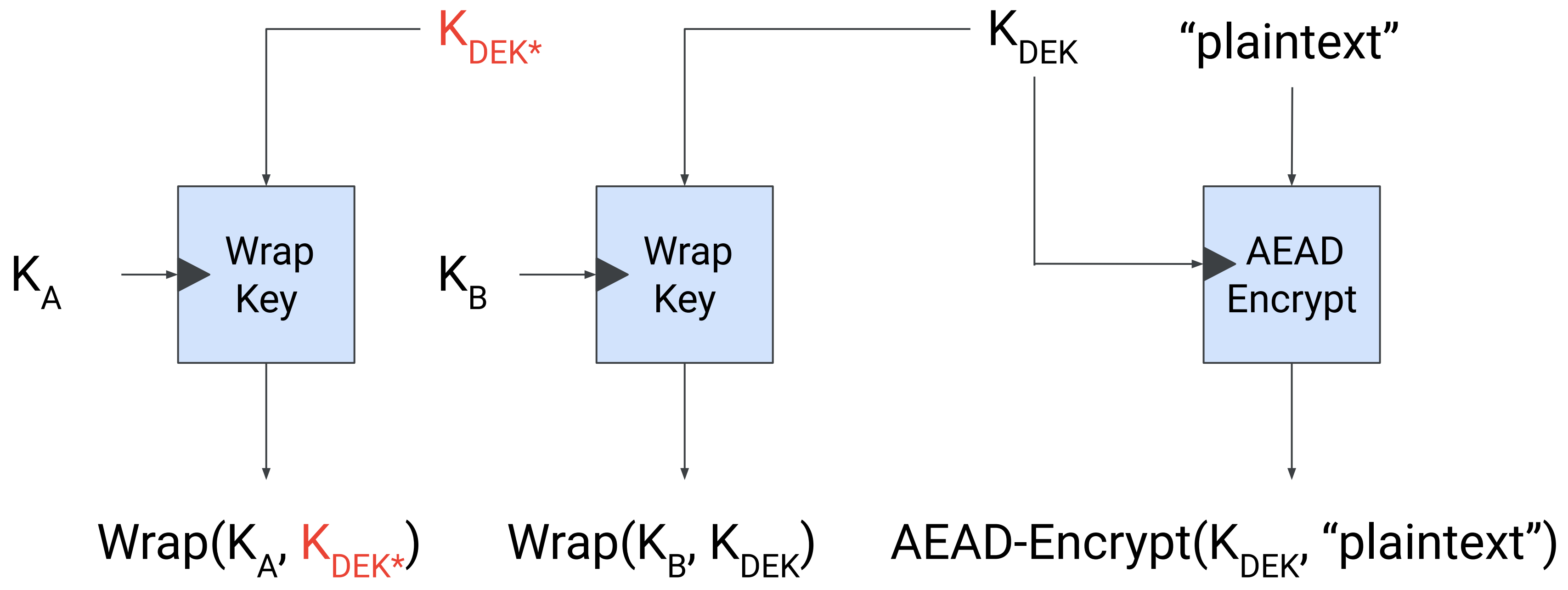
Case Study: Envelope Encryption



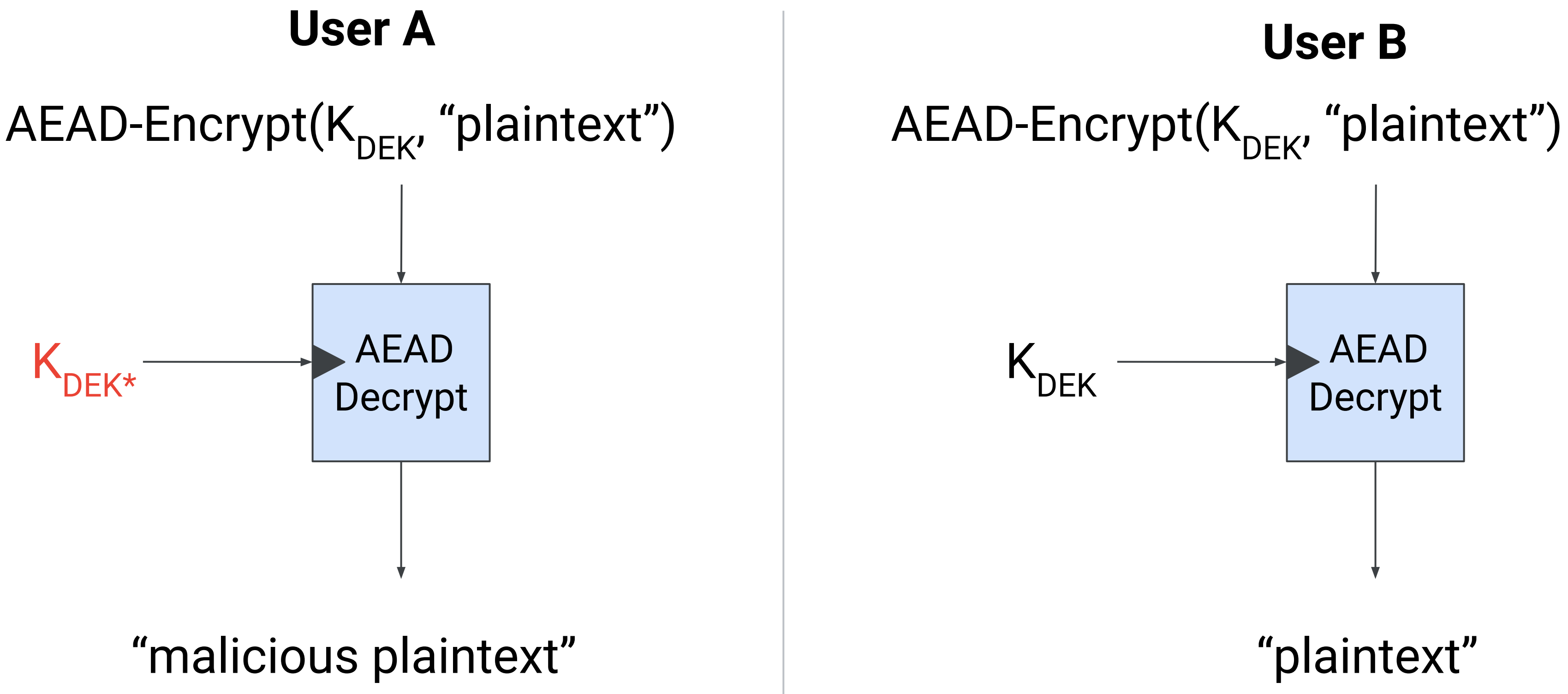
Case Study: Envelope Encryption



Case Study: Envelope Encryption



Case Study: Envelope Encryption



Case Study: Envelope Encryption

- Recipients receive **same** ciphertext.
- Might **falsely** assume that everyone decrypts to the same plaintext.
- Without a key-committing AEAD this is **not** true.
- AWS Encryption SDK was affected (< 2.0.0) and patched (CVE-2020-8897).

Practical Examples

- Key Rotation (see Paper)
- Subscribe with Google (see Paper)
- Facebook Message Franking (CRYPTO'18)
- Partitioning Oracle Attacks (USENIX'21)
- age file encryption (Mirco Stäuble, ETH Zurich)

Exploiting Lack of Key Commitment

Exploiting Lack of Key Commitment

Most commonly used AEADs are **not** key committing:

- AES-GCM, **AES-GCM-SIV**.
- ChaCha20-Poly1305.
- **OCB3**.

Exploiting Lack of Key Commitment

Constructing valid ciphertexts under multiple keys puts restrictions on plaintext:

- Include random blocks of data.
- Fixing bits in plaintext to specific values.

Exploiting Lack of Key Commitment

File formats have various restrictions:

- Starting sequences
- Headers
- Length fields
- …

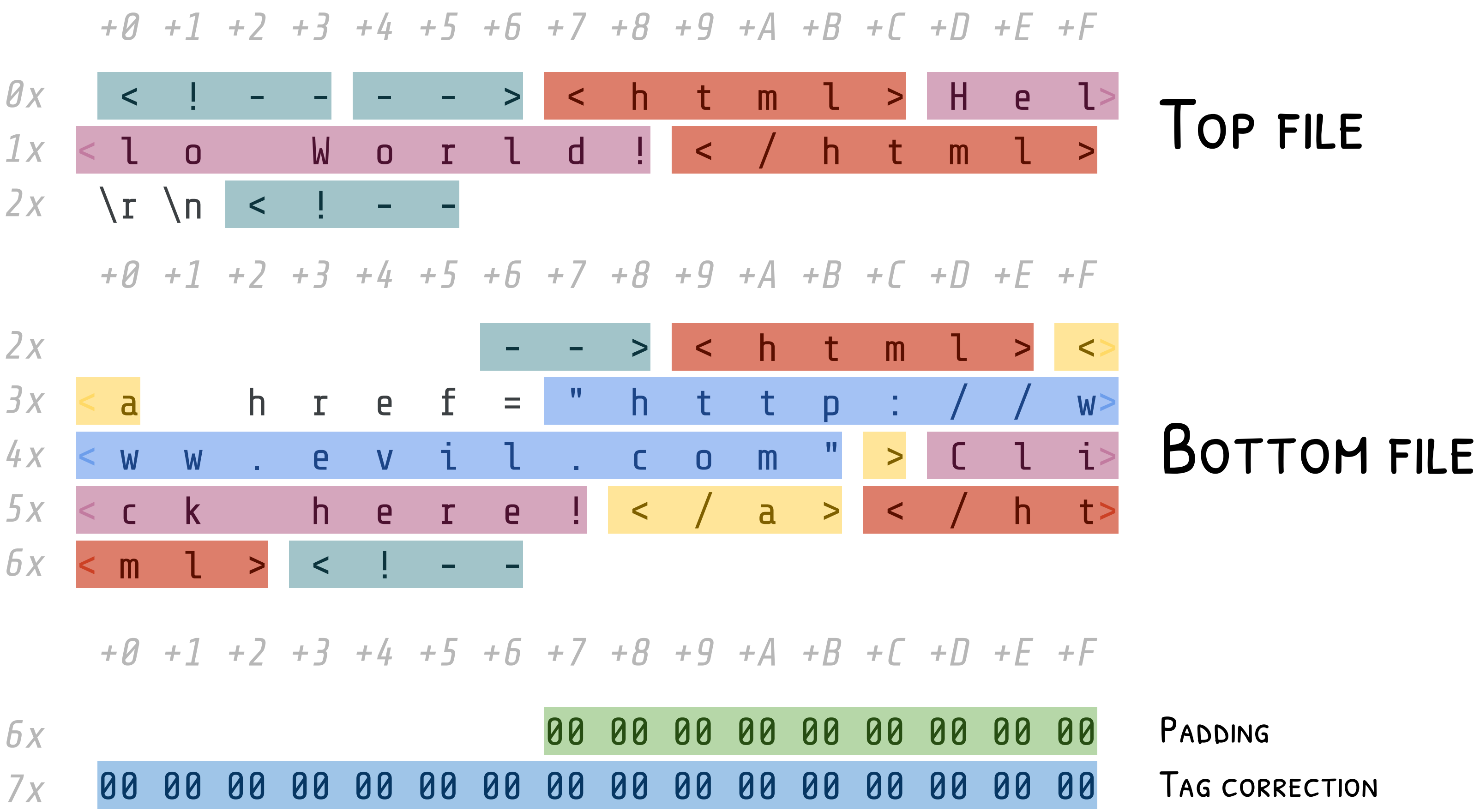
Exploiting Lack of Key Commitment

Can we still create meaningful plaintexts which are compliant with common file formats?

- Tooling supports 40+ formats, allows 270+ combinations, automated.
- Provide examples for PDF/PE, HTML/HTML…
- Our ePrint paper includes a PDF viewer :-)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x	<	!	-	-	-	-	>	<	h	t	m	l	>	H	e	l
1x	l	o		W	o	r	l	d	!	<	/	h	t	m	l	>
2x	\r	\n	<	!	-	-	-	-	>	<	h	t	m	l	>	<
3x	a		h	r	e	f	=	"	h	t	t	p	:	/	/	w
4x	w	w	.	e	v	i	l	.	c	o	m	"	>	C	l	i
5x	c	k		h	e	r	e	!	<	/	a	>	<	/	h	t
6x	m	l	>	<	!	-	-	00	00	00	00	00	00	00	00	00
7x	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F

Source file (combination of the original files)



+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
0x <!-- --> <html> Hel>
1x <lo World! </html>
2x \r \n <!--
Top file
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
2x --> <html> <>
3x
4x <ww.evil.com" > Cli>
5x <ck here! </ht>
6x <ml> <!--
Bottom file
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
6x 00 00 00 00 00 00 00 00 00
7x 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Padding
Tag correction

Dissection

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x	a2	ae	0f	b0	21	7b	96	71	6f	ff	96	73	4f	96	4a	9b
1x	b5	9f	0e	bd	c8	cd	2e	ab	9f	5f	4c	2b	1d	56	77	32
2x	c3	67	f7	35	0d	d4	75	a0	d5	be	e1	66	53	63	2b	eb
3x	24	34	ee	d2	da	23	70	66	ea	02	01	e8	b2	45	98	e0
4x	7f	b8	0b	ef	f3	91	eb	5c	7a	21	52	f8	71	7a	80	8f
5x	a5	41	82	b2	7e	43	b3	e3	13	09	9a	a9	b9	d8	71	81
6x	41	48	d0	ab	90	5f	6e	d4	2d	59	0d	a4	24	54	ac	f9
7x	85	39	a5	af	35	be	2c	db	dc	c1	07	bf	98	ce	0a	aa

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F

Ambiguous ciphertext

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x	7c	fa	a9	b5	-	-	>	<	h	t	m	l	>	H	e	l
1x	l	o		W	o	r	l	d	!	<	/	h	t	m	l	>
2x	\r	\n	<	!	-	-	db	15	fd	8a	87	3d	fb	47	11	d6
3x	28	37	f6	85	67	72	cb	13	24	6c	30	52	40	1e	d7	d9
4x	01	c4	21	a9	03	f5	ca	96	b3	58	eb	be	a5	6e	84	62
5x	30	a6	11	ea	a6	d8	0d	df	52	e5	34	76	65	7c	c3	31
6x	ce	5b	68	cf	a8	8c	33	a6	8d	e2	f8	8c	19	97	c0	3f
7x	f0	1f	4a	39	16	20	3d	bb	aa	9b	48	22	2a	2d	f4	a1

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F

First plaintext

Prefix (commented out)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x	7c	fa	a9	b5												

Top file

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x					-	-	>	<	h	t	m	l	>	H	e	l
1x	l	o		W	o	r	l	d	!	<	/	h	t	m	l	>
2x	\r	\n	<	!	-	-										

Suffix (commented out)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
2x							db	15	fd	8a	87	3d	fb	47	11	d6
3x	28	37	f6	85	67	72	cb	13	24	6c	30	52	40	1e	d7	d9
4x	01	c4	21	a9	03	f5	ca	96	b3	58	eb	be	a5	6e	84	62
5x	30	a6	11	ea	a6	d8	0d	df	52	e5	34	76	65	7c	c3	31
6x	ce	5b	68	cf	a8	8c	33	a6	8d	e2	f8	8c	19	97	c0	3f
7x	f0	1f	4a	39	16	20	3d	bb	aa	9b	48	22	2a	2d	f4	a1

Dissection of the first plaintext

Google

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x	3c	21	2d	2d	e4	04	01	d3	d0	c2	85	fb	e1	66	15	ea
1x	7c	5d	32	e7	bd	42	56	80	d1	0a	7d	5e	88	be	24	ad
2x	2e	f2	a0	a5	00	9a	-	-	>	<	h	t	m	l	>	<
3x	a		h	r	e	f	=	"	h	t	t	p	:	/	/	w
4x	w	w	.	e	v	i	l	.	c	o	m	"	>	C	l	i
5x	c	k		h	e	r	e	!	<	/	a	>	<	/	h	t
6x	m	l	>	<	!	-	-	00	00	00	00	00	00	00	00	00
7x	e3	aa	f1	16	df	ff	f4	55	83	8c	fa	8d	c5	17	70	e7
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F

Second plaintext

Prefix (commented out)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x	3c	21	2d	2d	e4	04	01	d3	d0	c2	85	fb	e1	66	15	ea
1x	7c	5d	32	e7	bd	42	56	80	d1	0a	7d	5e	88	be	24	ad
2x	2e	f2	a0	a5	00	9a										

Bottom file

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
2x							-	-	>	<	h	t	m	l	>	<
3x	a		h	r	e	f	=	"	h	t	t	p	:	/	/	w
4x	w	w	.	e	v	i	l	.	c	o	m	"	>	C	l	i
5x	c	k		h	e	r	e	!	<	/	a	>	<	/	h	t
6x	m	l	>	<	!	-	-									

Suffix (commented out)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
6x								00	00	00	00	00	00	00	00	00
7x	e3	aa	f1	16	df	ff	f4	55	83	8c	fa	8d	c5	17	70	e7

Dissection of the second plaintext

Adding Key Commitment

Adding Key Commitment

How to address lack of key commitment?

- Use key committing scheme in the first place.
- In paper we analyze two solutions compatible with AEADs:
    - Padding fix
    - Generic Construction
- Efficient Schemes for Committing Authenticated Encryption (EUROCRYPT'22).

Conclusion

Takeaways:

- Lack of key commitment is an issue in real-world applications.
- AEADs should be explicit about providing this property or not.

Resources available:

- https://eprint.iacr.org/2020/1456
- https://github.com/corkami/mitra
- https://github.com/kste/keycommitment

Questions?