Cryptanalysis of OCB2: Attacks on Authenticity and Confidentiality.

Based on

- Akiko Inoue, Tetsu Iwata, Kazuhiko Minematsu, and Bertram Poettering
- Cryptanalysis of OCB2: Attacks on Authenticity and Confidentiality, CRYPTO 2019
- Cryptology ePrint Archive: Report 2019/311

Penguin

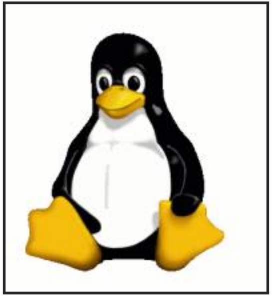
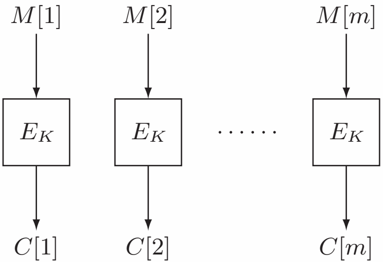


Image: Larry Ewing, lewing@isc.tamu.edu, created with the GIMP. https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

ECB (Electronic Code Book)



- E_K: a block cipher with n-bit blocks
- $M = (M[1], \ldots, M[m])$
- $C = (C[1], \ldots, C[m])$

The ECB Penguin

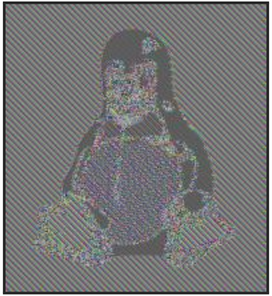
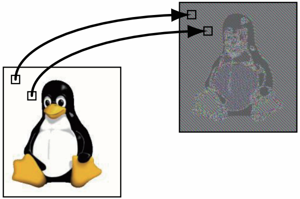


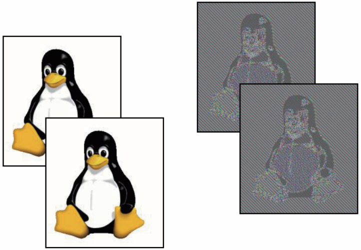
Image: Larry Ewing, lewing@isc.tamu.edu, created with the GIMP. https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Issues with ECB

- $M[i] = M[j] \Rightarrow C[i] = C[j]$



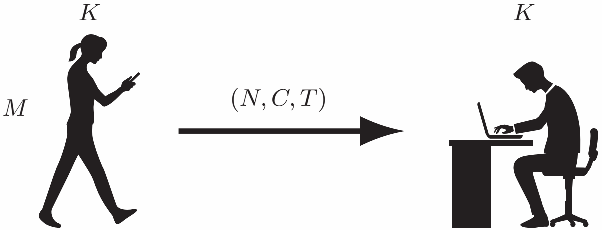
- $M = M' \Rightarrow C = C'$



- does not provide authenticity, “authenticated encryption”

AE (Authenticated Encryption)

- Symmetric-key primitive for privacy and authenticity
- Nonce-based AE [Rog02] (this talk will not consider associated data)
  - nonce: data that is changed for each encryption (counter)
- Encryption: $(K, N, M) \mapsto (C, T)$
- Decryption: $(K, N, C, T) \mapsto M$ or $(K, N, C, T) \mapsto \perp$ (authentication error, reject)



K: key
N: nonce
M: message
C: ciphertext
T: tag

$(C, T) \leftarrow \mathrm{Enc}_K(N, M)$

$M/\perp \leftarrow \mathrm{Dec}_K(N, C, T)$

[Rog02] Rogaway. Authenticated-encryption with associated-data. CCS 2002

Examples of AE

- GCM and CCM (NIST recommendations)
- 6 schemes in ISO/IEC 19772
- IETF RFC includes GCM, ChaCha20-Poly1305, ...
- 6 schemes in CAESAR final portfolio
- many schemes in the ongoing NIST lightweight cryptography standardization project

OCB (Offset Code Book)

- 3 versions, built on a block cipher (e.g., AES, with $n = 128$)
  - OCB1 by Rogaway et al. at CCS 2001 [RBBK01]
  - OCB2 by Rogaway at ASIACRYPT 2004 [Rog04]
  - OCB3 by Krovetz and Rogaway at FSE 2011 [KR11]
- Nonce-based AE (with AD) with strong features:
  - fully parallelizable
  - 1 block cipher call to process each n-bit block (rate-1, same as CTR and ECB modes)
  - provable security

[RBBK01] Rogaway, Bellare, Black, Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. CCS 2001

[Rog04] Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. ASIACRYPT 2004

[KR11] Krovetz, Rogaway. The software performance of authenticated-encryption modes. FSE 2011

Security Evaluation of OCB

All versions have been extensively studied:

- Security proofs for all versions of OCB by Rogaway et al. [RBBK01, Rog04, KR11]
- Tightness of the security bounds: Ferguson [Fer02], Sun et al. [SWZ12]
- (Nonce) misuse attacks: Andreeva et al. [ABLMMY14], Ashur et al. [ADL17]
- Necessity of SPRP: Aoki and Yasuda [AY13]
- Bound improvement (for OCB3): Bhaumik and Nandi [BN17]

[Fer02] Ferguson. Collision attacks on OCB. Comments to NIST, 2002

[SWZ12] Sun, Wang, Zhang. Collision attacks on variant of OCB mode and its series. Inscrypt 2012

[ABLMMY14] Andreeva, Bogdanov, Luykx, Mennink, Mouha, Yasuda. How to securely release unverified plaintext in authenticated encryption. ASIACRYPT 2014

[ADL17] Ashur, Dunkelman, Luykx. Boosting authenticated encryption robustness with minimal modifications. CRYPTO 2017

[AY13] Aoki, Yasuda. The security of the OCB mode of operation without the SPRP assumption. ProvSec 2013

[BN17] Bhaumik, Nandi. Improved security for OCB3. ASIACRYPT 2017

Security Evaluation of OCB

All versions have been extensively studied:

- Security proofs for all versions of OCB by Rogaway et al. [RBBK01, Rog04, KR11]
- Tightness of the security bounds: Ferguson [Fer02], Sun et al. [SWZ12]
- (Nonce) misuse attacks: Andreeva et al. [ABLMMY14], Ashur et al. [ADL17]
- Necessity of SPRP: Aoki and Yasuda [AY13]
- Bound improvement (for OCB3): Bhaumik and Nandi [BN17]

No weakness known, the security is very well understood

[Fer02] Ferguson. Collision attacks on OCB. Comments to NIST, 2002

[SWZ12] Sun, Wang, Zhang. Collision attacks on variant of OCB mode and its series. Inscrypt 2012

[ABLMMY14] Andreeva, Bogdanov, Luykx, Mennink, Mouha, Yasuda. How to securely release unverified plaintext in authenticated encryption. ASIACRYPT 2014

[ADL17] Ashur, Dunkelman, Luykx. Boosting authenticated encryption robustness with minimal modifications. CRYPTO 2017

[AY13] Aoki, Yasuda. The security of the OCB mode of operation without the SPRP assumption. ProvSec 2013

[BN17] Bhaumik, Nandi. Improved security for OCB3. ASIACRYPT 2017

Security Evaluation of OCB

All versions have been extensively studied:

- Security proofs for all versions of OCB by Rogaway et al. [RBBK01, Rog04, KR11]
- Tightness of the security bounds: Ferguson [Fer02], Sun et al. [SWZ12]
- (Nonce) misuse attacks: Andreeva et al. [ABLMMY14], Ashur et al. [ADL17]
- Necessity of SPRP: Aoki and Yasuda [AY13]
- Bound improvement (for OCB3): Bhaumik and Nandi [BN17]

No weakness known, the security is very well understood?

[Fer02] Ferguson. Collision attacks on OCB. Comments to NIST, 2002

[SWZ12] Sun, Wang, Zhang. Collision attacks on variant of OCB mode and its series. Inscrypt 2012

[ABLMMY14] Andreeva, Bogdanov, Luykx, Mennink, Mouha, Yasuda. How to securely release unverified plaintext in authenticated encryption. ASIACRYPT 2014

[ADL17] Ashur, Dunkelman, Luykx. Boosting authenticated encryption robustness with minimal modifications. CRYPTO 2017

[AY13] Aoki, Yasuda. The security of the OCB mode of operation without the SPRP assumption. ProvSec 2013

[BN17] Bhaumik, Nandi. Improved security for OCB3. ASIACRYPT 2017

Our Results

Structural weakness of OCB2

- Independent of the underlying block cipher (and its block size)
- has been overlooked for about 15 years

Attacks

- Authenticity attacks (existential and universal forgeries)
- Privacy attacks (distinguishing attack and plaintext recovery)
- All attacks have very small complexity & the success probability is (almost) one

Practical Impacts

- OCB2 was one of the six algorithms in ISO/IEC 19772
  - ISO/IEC declared a plan for removal of OCB2 from the international standard
- SJCL Javascript crypto library implements OCB2
  - Users may be affected, though it is hard to see the real impact
  - Fixing crypto is not easy, time-consuming
- Joplin, a multi-platform application for taking notes
  - uses OCB2 through SJCL
  - decided to wait for the decision by SJCL team

---

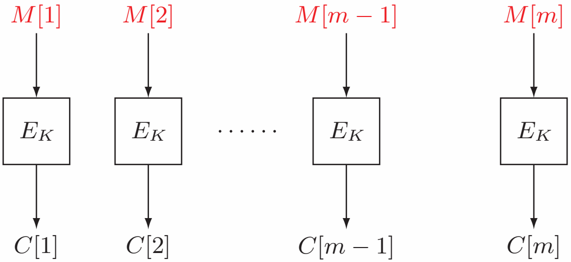
http://bitwiseshiftleft.github.io/sjcl/

https://joplinapp.org/

https://github.com/laurent22/joplin/issues/943

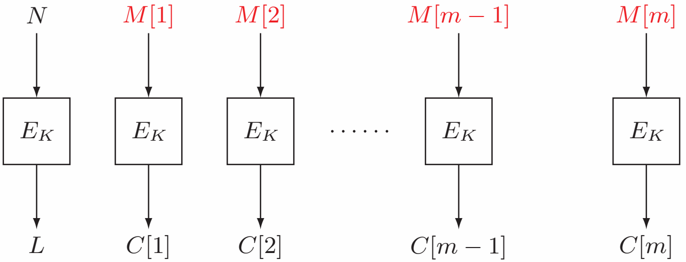
Technical Details of OCB2

- Encryption: $(N, M) \mapsto (C, T)$, ECB mode with masks generated from $L = E_K(N)$
  - $2a$ is doubling of a over $\mathrm{GF}(2^n)$, $3a = 2a \oplus a$
  - $M[m]$ is encrypted in CTR mode
  - $\texttt{len}(X)$ is an n-bit encoding of $|X|$
- The checksum is $\Sigma = M[1] \oplus \cdots \oplus M[m]$



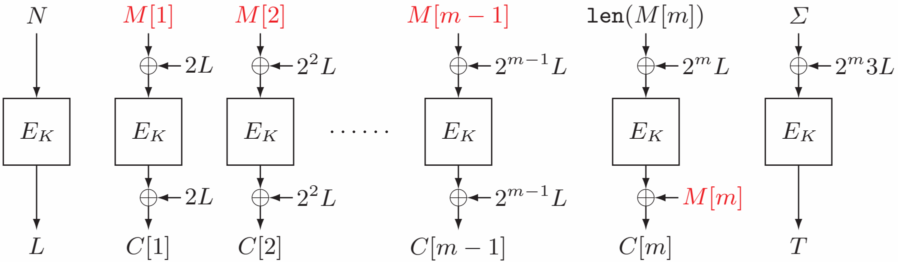
Technical Details of OCB2

- Encryption: $(N, M) \mapsto (C, T)$, ECB mode with masks generated from $L = E_K(N)$
  - $2a$ is doubling of a over $\mathrm{GF}(2^n)$, $3a = 2a \oplus a$
  - $M[m]$ is encrypted in CTR mode
  - $\mathtt{len}(X)$ is an n-bit encoding of $|X|$
- The checksum is $\Sigma = M[1] \oplus \cdots \oplus M[m]$



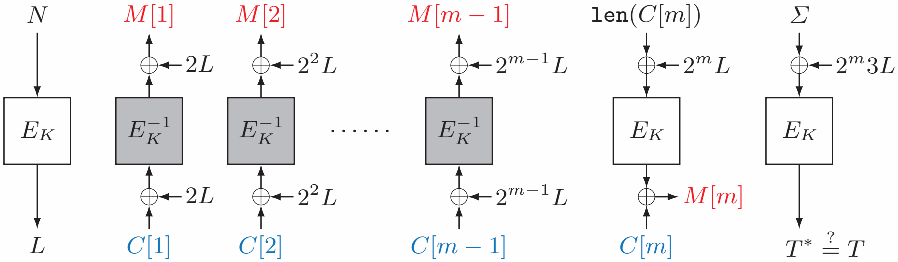
Technical Details of OCB2

- Encryption: $(N, M) \mapsto (C, T)$, ECB mode with masks generated from $L = E_K(N)$
  - $2a$ is doubling of a over $\mathrm{GF}(2^n)$, $3a = 2a \oplus a$
  - $M[m]$ is encrypted in CTR mode
  - $\texttt{len}(X)$ is an n-bit encoding of $|X|$
- The checksum is $\Sigma = M[1] \oplus \cdots \oplus M[m]$



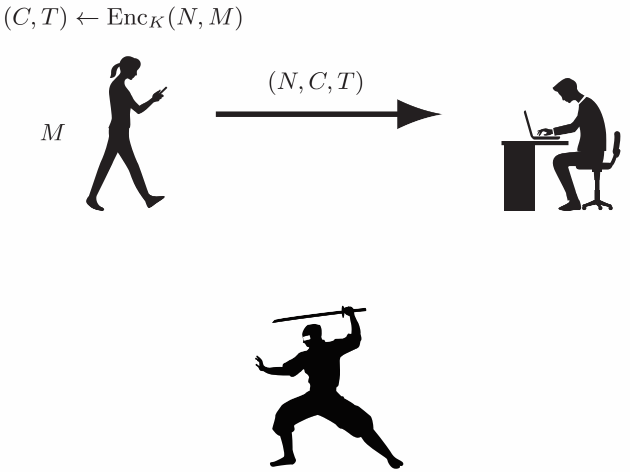
Technical Details of OCB2

- Decryption: $(N, C, T) \mapsto M/\perp$



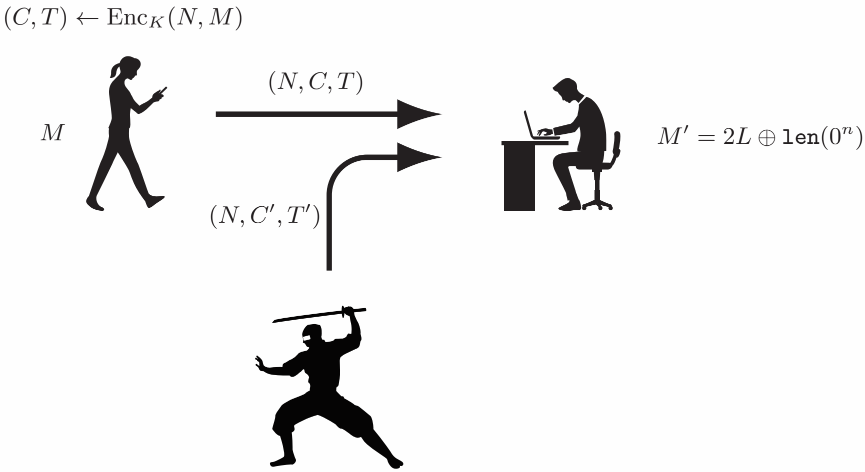
- Theorem [Rog04]
  - $(C, T) \approx \text{random string}$ (privacy)
  - forgery is not possible (authenticity)

Minimal Forgery [IM18] (Existential Forgery)



[IM18] Inoue, Minematsu. Cryptanalysis of OCB2. IACR ePrint 2018/1040

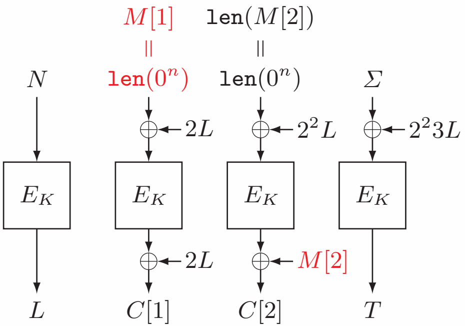
Minimal Forgery [IM18] (Existential Forgery)



[IM18] Inoue, Minematsu. Cryptanalysis of OCB2. IACR ePrint 2018/1040

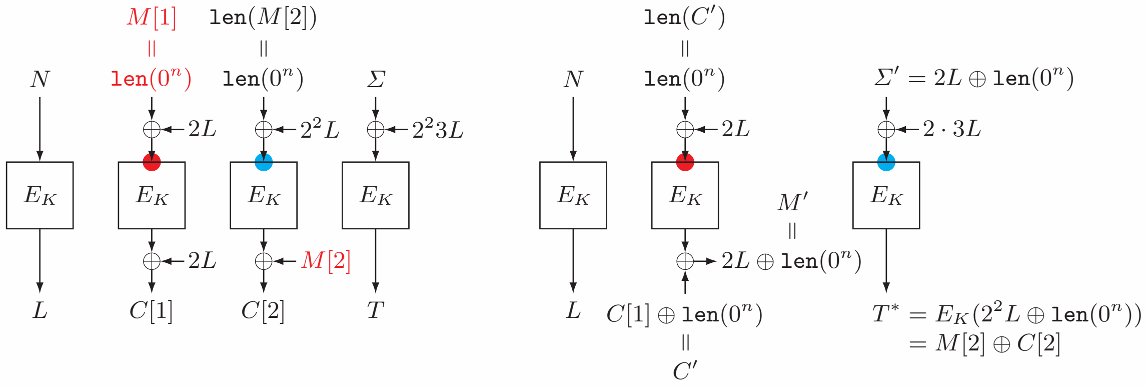
Minimal Forgery [IM18] (For Experts)

- Encrypt (N, M) to obtain $(C[1], C[2], T)$, where $M = (\texttt{len}(0^n), M[2])$ and $|M[2]| = n$



Minimal Forgery [IM18] (For Experts)

- Encrypt (N, M) to obtain $(C[1], C[2], T)$, where $M = (\texttt{len}(0^n), M[2])$ and $|M[2]| = n$
- Decrypt (N, C', T'), where $C' = C[1] \oplus \texttt{len}(0^n)$ and $T' = M[2] \oplus C[2]$
  - Note: $2L \oplus 2 \cdot 3L = 2L \oplus 2(2+1)L = 2^2 L$
  - $M' = 2L \oplus \texttt{len}(0^n)$ is returned, L can be used for powerful attacks



Attacks

Quickly triggered other attacks [IIMP19] (all under CCA):

- forgery of longer messages
- universal forgery
- distinguishing attack
- plaintext recovery
- simulation of block cipher encryption
- simulation of block cipher decryption

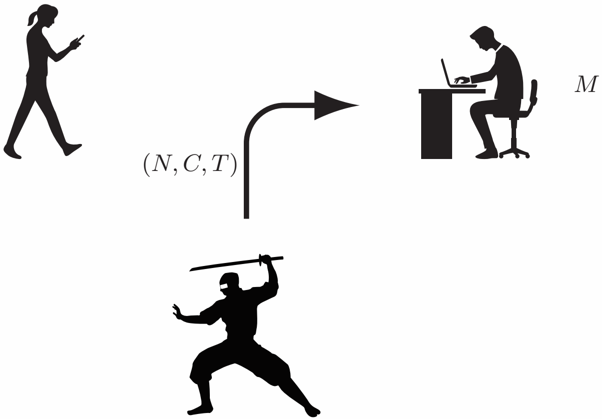
[IIMP19] Inoue, Iwata, Minematsu, Poettering. Cryptanalysis of OCB2: Attacks on Authenticity and Confidentiality. CRYPTO 2019

Universal Forgery [IIMP19]

- "Minimal forgery" forges $M' = 2L \oplus \texttt{len}(0^n)$
- Universal forgery: for any (N, M), the adversary can compute (C, T)

Universal Forgery [IIMP19]

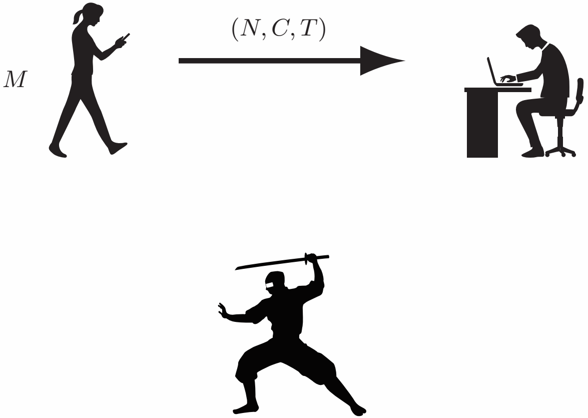
- "Minimal forgery" forges $M' = 2L \oplus \texttt{len}(0^n)$
- Universal forgery: for any (N, M), the adversary can compute (C, T)
  - uses the minimal forgery as a subroutine, the most powerful authenticity attack



Plaintext Recovery [IIMP19]

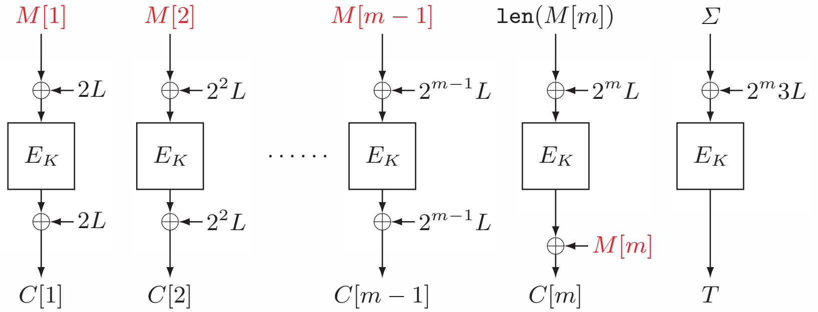
- The most powerful privacy attack: for any (N, C, T) (for unknown M), the adversary can compute M
  - uses the minimal forgery as a subroutine

$$(C, T) \leftarrow \mathrm{Enc}_K(N, M)$$



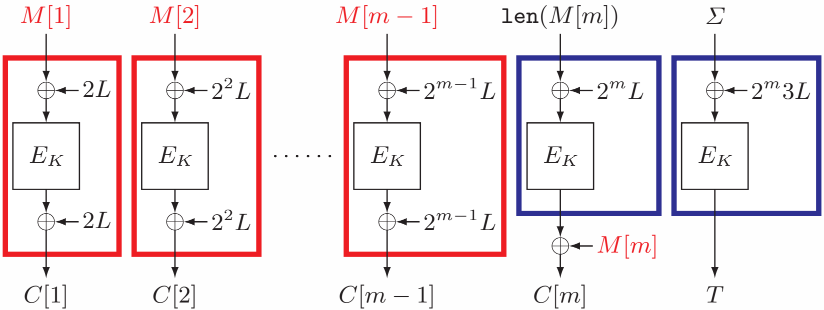
What Went Wrong?

- OCB2 has a proof of security [Rog04]
- The technical specification of OCB2 is too complex for a direct security proof
- To prove the security, [Rog04] uses "abstraction" of OCB2

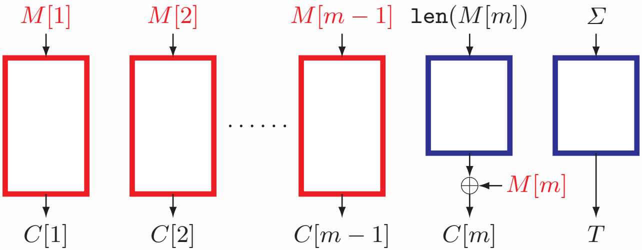


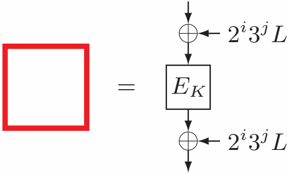
What Went Wrong?

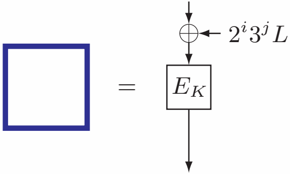
- OCB2 has a proof of security [Rog04]
- The technical specification of OCB2 is too complex for a direct security proof
- To prove the security, [Rog04] uses "abstraction" of OCB2



What Went Wrong?

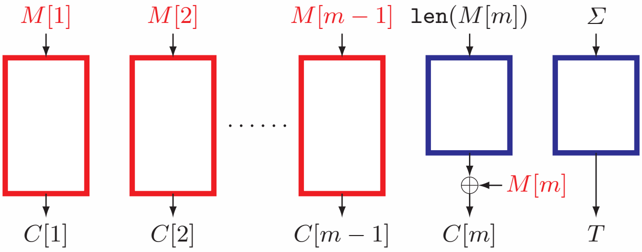


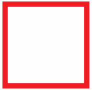
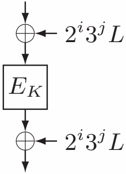
- If 

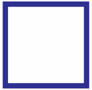
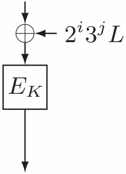
(XEX) and 

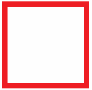
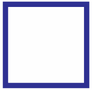
(XE), then OCB2

What Went Wrong?



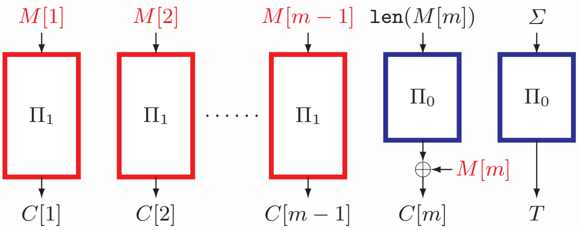
- If  $=$ 

(XEX) and  $=$ 

(XE), then OCB2
- If  is Π_1 (ideally secure XEX) and  is Π_0 (ideally secure XE), then $\Theta\mathrm{CB2}$

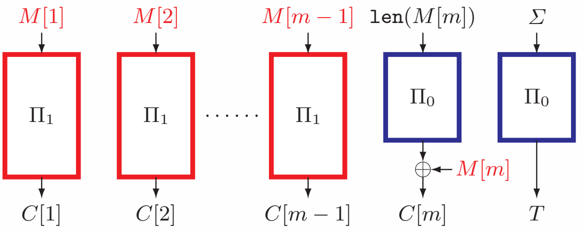
What Went Wrong?

- Three steps to prove the security of OCB2
  - Prove that $\Theta\mathrm{CB2}$ is secure (privacy and authenticity)
  - Prove that for any "tag-respecting" adversary, $(\mathsf{XEX}, \mathsf{XE}) \approx (\Pi_1, \Pi_0)$
  - Conclude that OCB2 is secure, "hybrid argument"



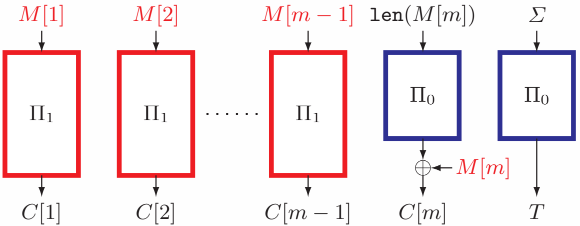
What Went Wrong?

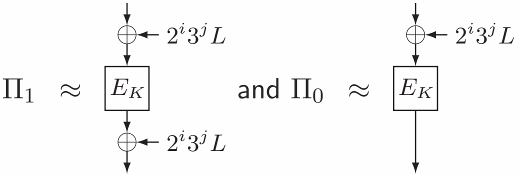
- Three steps to prove the security of OCB2
  - Prove that $\Theta\mathrm{CB2}$ is secure (privacy and authenticity) ✓
  - Prove that for any "tag-respecting" adversary, $(\mathsf{XEX}, \mathsf{XE}) \approx (\Pi_1, \Pi_0)$
  - Conclude that OCB2 is secure, "hybrid argument"



What Went Wrong?

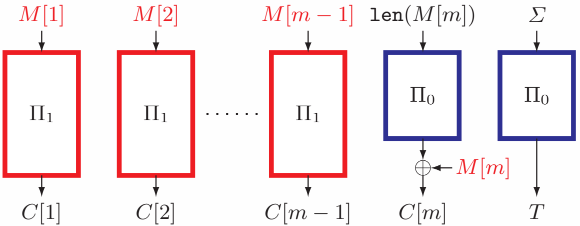
- Three steps to prove the security of OCB2
  - Prove that $\Theta\mathrm{CB2}$ is secure (privacy and authenticity) ✓
  - Prove that for any "tag-respecting" adversary, $(\mathsf{XEX}, \mathsf{XE}) \approx (\Pi_1, \Pi_0)$
  - Conclude that OCB2 is secure, "hybrid argument"

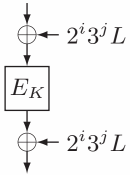


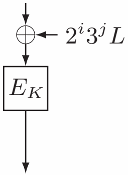


What Went Wrong?

- Three steps to prove the security of OCB2
  - Prove that $\Theta\mathrm{CB2}$ is secure (privacy and authenticity) ✓
  - Prove that for any "tag-respecting" adversary, $(\mathsf{XEX}, \mathsf{XE}) \approx (\Pi_1, \Pi_0)$ ✓
  - Conclude that OCB2 is secure, "hybrid argument"

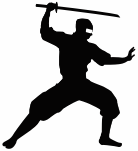


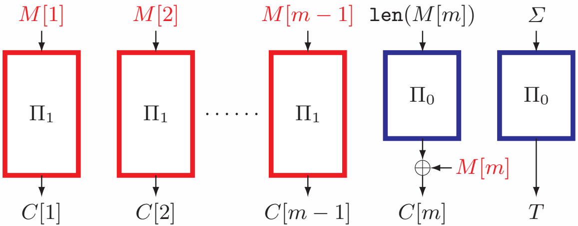
$\Pi_1 \approx$ 

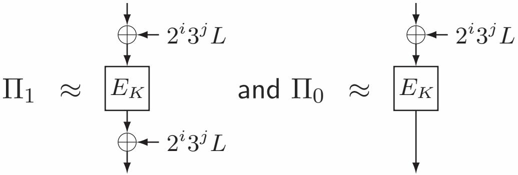
and $\Pi_0 \approx$ 

What Went Wrong?

- Three steps to prove the security of OCB2
  - Prove that $\Theta\mathrm{CB2}$ is secure (privacy and authenticity) ✓
  - Prove that for any "tag-respecting" adversary, $(\mathsf{XEX}, \mathsf{XE}) \approx (\Pi_1, \Pi_0)$ ✓
  - Conclude that OCB2 is secure, "hybrid argument"

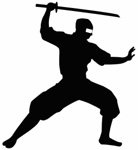


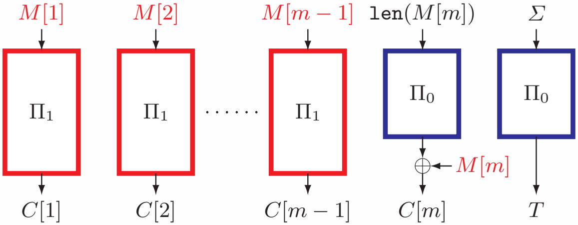


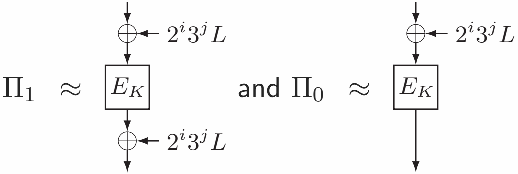


What Went Wrong?

- Three steps to prove the security of OCB2
  - Prove that $\Theta\mathrm{CB2}$ is secure (privacy and authenticity) ✓
  - Prove that for any "tag-respecting" adversary, $(\mathsf{XEX}, \mathsf{XE}) \approx (\Pi_1, \Pi_0)$ ✓
  - Conclude that OCB2 is secure, "hybrid argument" — XEX/XE misused, does not work!







What Went Wrong?

- Three steps to prove the security of OCB2
  - Prove that $\Theta\mathrm{CB2}$ is secure (privacy and authenticity) ✓
  - Prove that for any tag-respecting adversary, $(\mathsf{XEX}, \mathsf{XE}) \approx (\Pi_1, \Pi_0)$ ✓
  - Conclude that OCB2 is secure, "hybrid argument" — XEX/XE misused, does not work!
- Lesson learned: prove all the statements PLUS carefully check they fit together

Fixes

Some ways to fix OCB2

- Use XEX for the last message block (OCB2f, provably secure)
- Change the definition of the mask (OCB2ff, provably secure, will be included in [ePrint 2019/311])
- Other potential (unproven) options
  - Always-nonempty-AD, always-PMAC
  - Rejecting harmful inputs to OCB2 (Counter-cryptanalysis)

Some ways to avoid OCB2

- GCM
- OCB3

Timeline

2001	OCB1 proposed at CCS 2001
2004	OCB2 proposed at ASIACRYPT 2004
2009	OCB2 included in ISO/IEC 19772:2009
2011	OCB3 proposed at FSE 2011
2014	OCB3 in IETF RFC 7253

Timeline

2001	OCB1 proposed at CCS 2001
2004	OCB2 proposed at ASIACRYPT 2004
2009	OCB2 included in ISO/IEC 19772:2009
2011	OCB3 proposed at FSE 2011
2014	OCB3 in IETF RFC 7253
2018 Sep 06	A potential gap in the proof found by IM

Timeline

2001	OCB1 proposed at CCS 2001
2004	OCB2 proposed at ASIACRYPT 2004
2009	OCB2 included in ISO/IEC 19772:2009
2011	OCB3 proposed at FSE 2011
2014	OCB3 in IETF RFC 7253
2018 Sep 06	A potential gap in the proof found by IM
2018 Oct 09	Minimal forgery found by IM

Timeline

2001	OCB1 proposed at CCS 2001
2004	OCB2 proposed at ASIACRYPT 2004
2009	OCB2 included in ISO/IEC 19772:2009
2011	OCB3 proposed at FSE 2011
2014	OCB3 in IETF RFC 7253
2018 Sep 06	A potential gap in the proof found by IM
2018 Oct 09	Minimal forgery found by IM
2018 Oct 26	[IM18] sent to ePrint, minimal forgery, extension to longer messages, almost universal forgery, gap, fix

Timeline

2001	OCB1 proposed at CCS 2001
2004	OCB2 proposed at ASIACRYPT 2004
2009	OCB2 included in ISO/IEC 19772:2009
2011	OCB3 proposed at FSE 2011
2014	OCB3 in IETF RFC 7253
2018 Sep 06	A potential gap in the proof found by IM
2018 Oct 09	Minimal forgery found by IM
2018 Oct 26	[IM18] sent to ePrint, minimal forgery, extension to longer messages, almost universal forgery, gap, fix
2018 Nov 08	[Poe18] sent to ePrint, distinguishing attack (posted on Nov 09, 16:00)

Timeline

2001	OCB1 proposed at CCS 2001
2004	OCB2 proposed at ASIACRYPT 2004
2009	OCB2 included in ISO/IEC 19772:2009
2011	OCB3 proposed at FSE 2011
2014	OCB3 in IETF RFC 7253
2018 Sep 06	A potential gap in the proof found by IM
2018 Oct 09	Minimal forgery found by IM
2018 Oct 26	[IM18] sent to ePrint, minimal forgery, extension to longer messages, almost universal forgery, gap, fix
2018 Nov 08	[Poe18] sent to ePrint, distinguishing attack (posted on Nov 09, 16:00)
2018 Nov 11 10:00	[Iwa18] sent to ePrint, full plaintext recovery (posted on Nov 12, 02:00)

Timeline

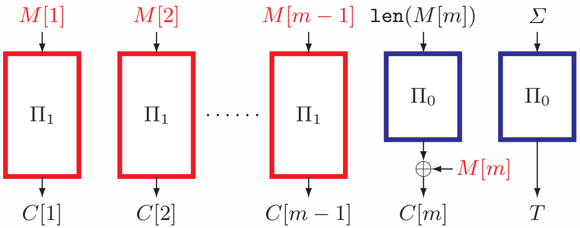
2001	OCB1 proposed at CCS 2001
2004	OCB2 proposed at ASIACRYPT 2004
2009	OCB2 included in ISO/IEC 19772:2009
2011	OCB3 proposed at FSE 2011
2014	OCB3 in IETF RFC 7253
2018 Sep 06	A potential gap in the proof found by IM
2018 Oct 09	Minimal forgery found by IM
2018 Oct 26	[IM18] sent to ePrint, minimal forgery, extension to longer messages, almost universal forgery, gap, fix
2018 Nov 08	[Poe18] sent to ePrint, distinguishing attack (posted on Nov 09, 16:00)
2018 Nov 11 10:00	[Iwa18] sent to ePrint, full plaintext recovery (posted on Nov 12, 02:00)
2018 Nov 11 14:00	[IM18] updated, universal forgery
2018 Nov 11 22:00	[Poe18] updated, BC simulation, universal forgery, partial plaintext recovery
2018 Nov 12 16:00	[Poe18] updated, full plaintext recovery
2018 Nov 16	[Iwa18] updated, BC decryption simulation

Timeline

- An exciting competition!
  - multiple teams from industry and academia (NEC, Nagoya U, and IBM & RHUL), across different corners in the world
- After the first finding of the potential gap, everything happened in a very short period of time
- Attacks only get better, for the case of OCB2, attacks got better very quickly

Concluding Remarks

- OCB2 is broken.
  - should not be used
- A (seemingly) small flaw in the proof led to surprisingly powerful attacks
- Not applicable to OCB1 and OCB3
  - They do not misuse XEX/XE
  - The general structure of OCB is sound

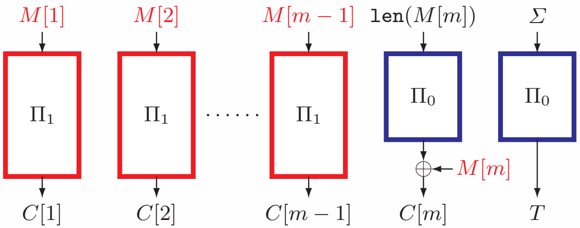


Lessons learned

- Even the most promising scheme can fail
- Active third-party verification of security proofs is important

Concluding Remarks

- OCB2 is broken.
  - should not be used
- A (seemingly) small flaw in the proof led to surprisingly powerful attacks
- Not applicable to OCB1 and OCB3
  - They do not misuse XEX/XE
  - The general structure of OCB is sound



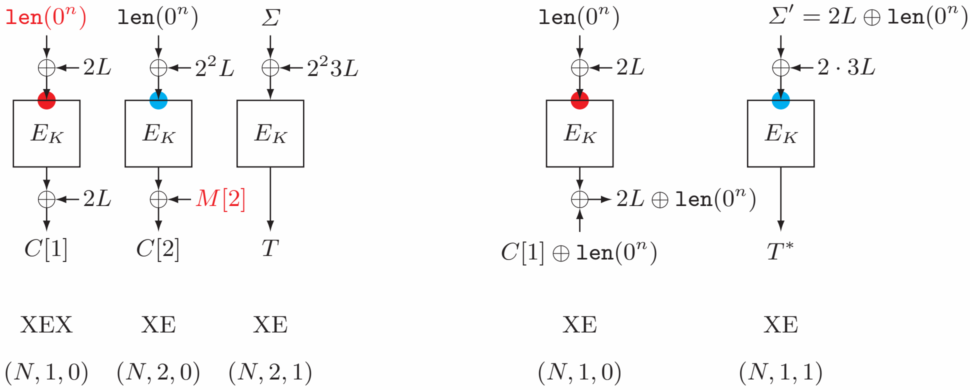
Lessons learned

- Even the most promising scheme can fail
- Active third-party verification of security proofs is important

Thanks!

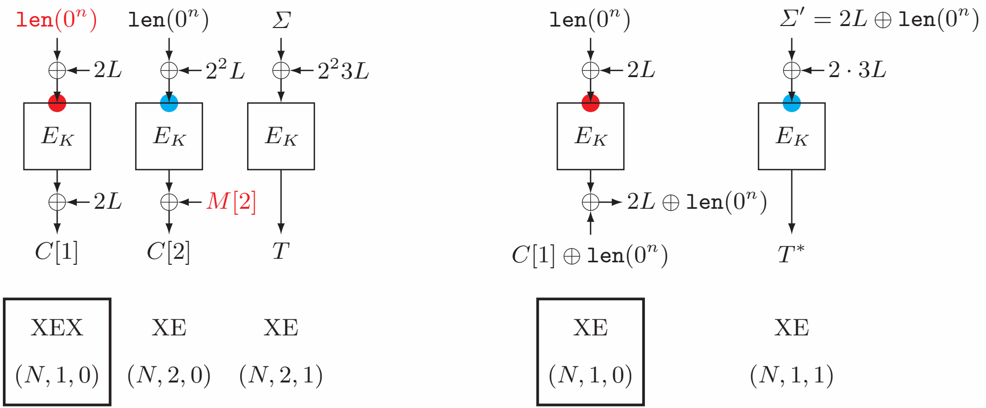
The Hybrid (For Experts, Backup Slide)

- If there is an adversary against OCB2, then there is an adversary against XEX/XE



The Hybrid (For Experts, Backup Slide)

- If there is an adversary against OCB2, then there is an adversary against XEX/XE



- Without violating the tag-respecting condition, the simulation is impossible. The hybrid does not work