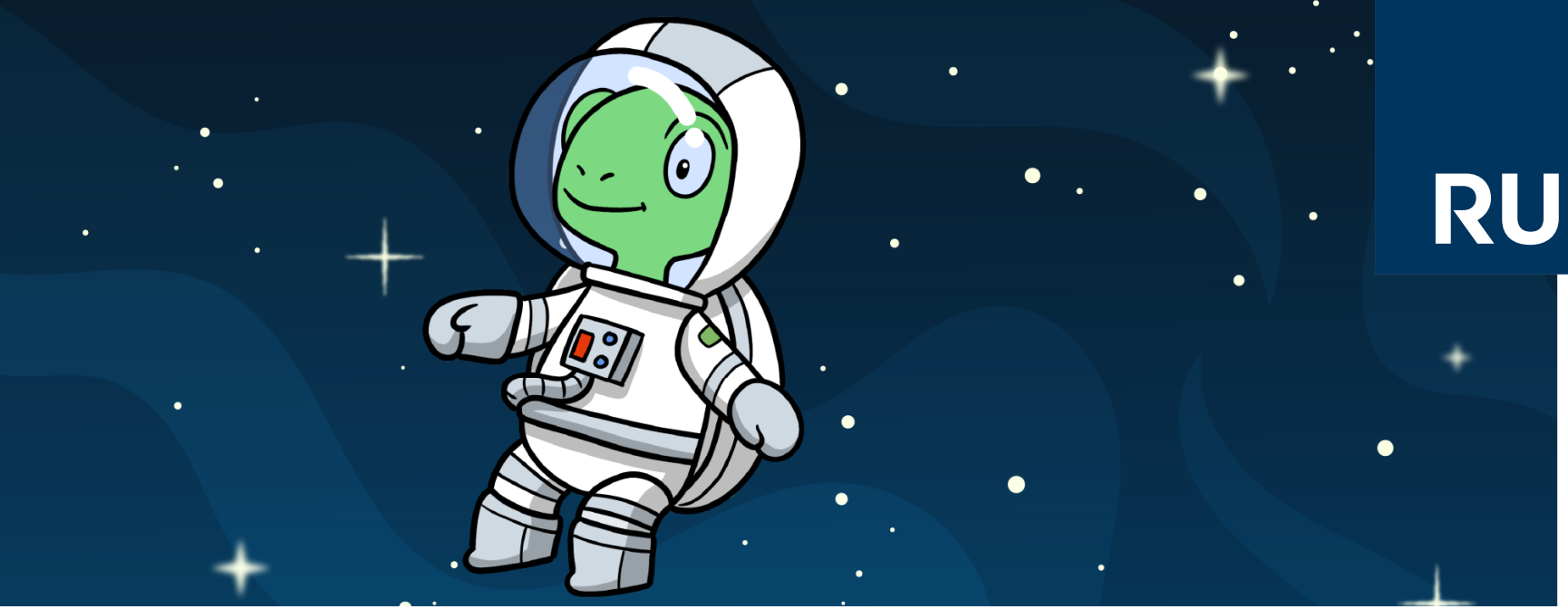


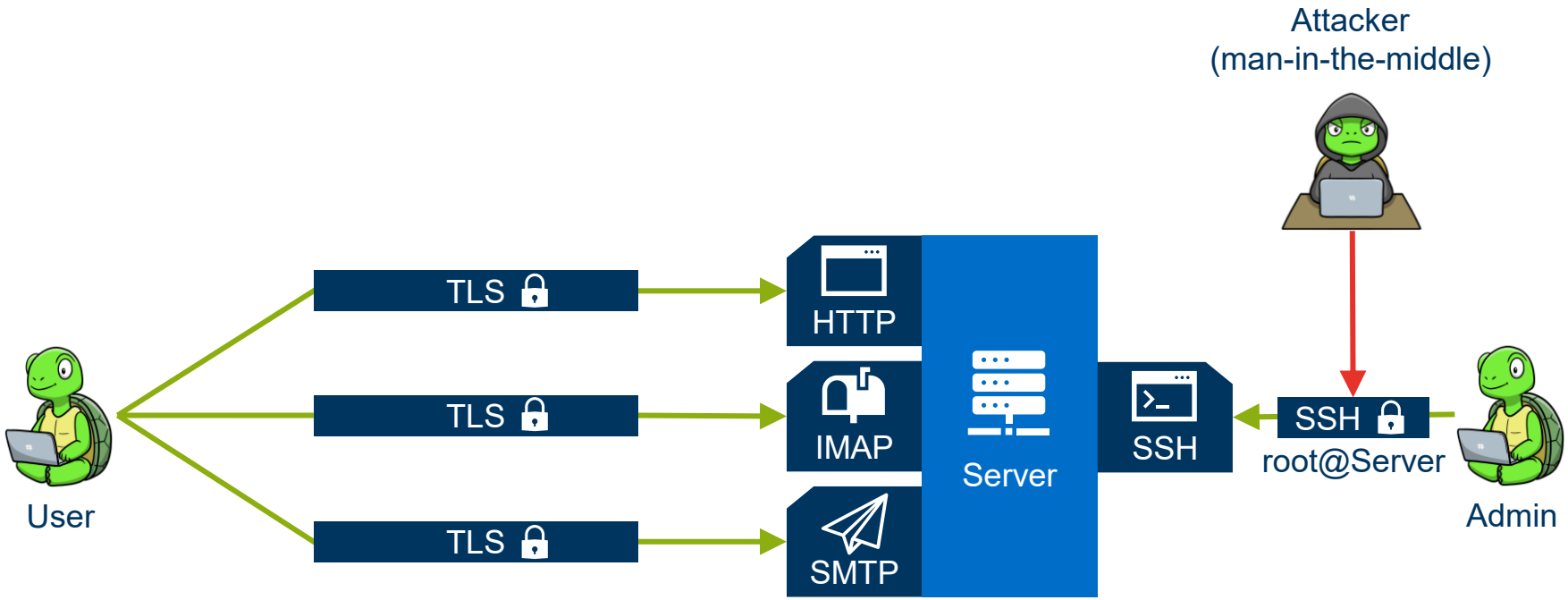
RUB

RUHR-UNIVERSITÄT BOCHUM

TERRAPIN ATTACK: BREAKING SSH CHANNEL INTEGRITY BY SEQUENCE NUMBER MANIPULATION

Fabian Bäumer, Marcus Brinkmann, Jörg Schwenk | Workshop on Attacks in Cryptography 7

SSH Is Often Used for High Privilege Server Access



SSH Is Split Into Separate Layers

SSH Connection Protocol [RFC4254]

SSH Authentication Protocol [RFC4252]

SSH Transport Layer Protocol (TLP) [RFC4253]

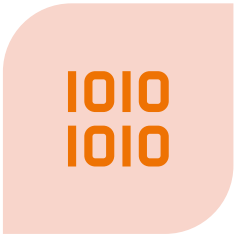
→ Binary Packet Protocol

→ SSH Handshake

TCP / IP

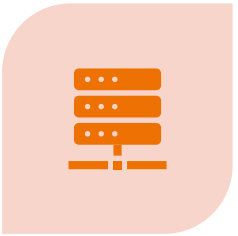
The SSH TLP Has Four Major Security Goals

CONFIDENTIALITY



INTEGRITY

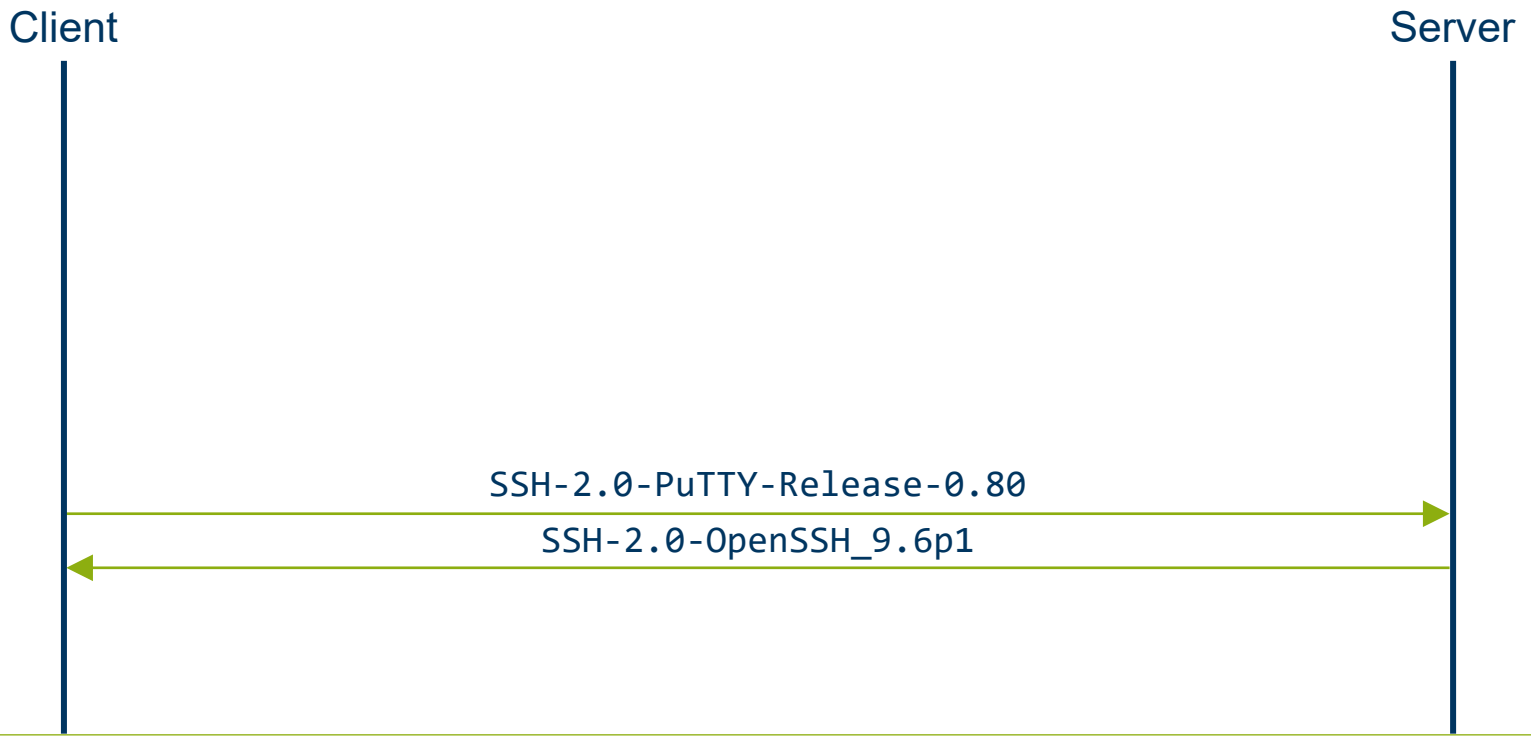
AUTHENTICITY



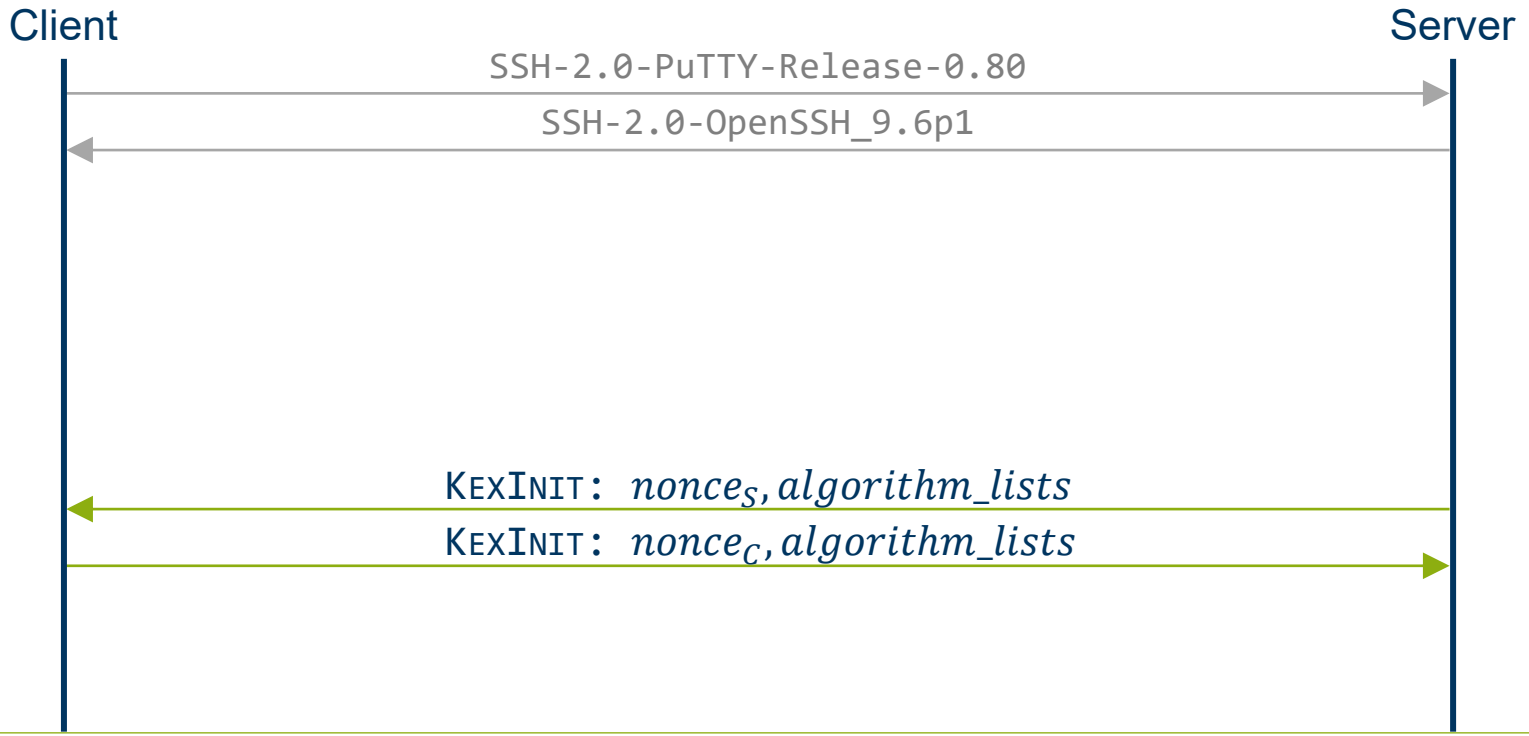
SERVER AUTHENTICATION

Our Finding: SSH fails to protect the integrity of the secure channel under certain conditions

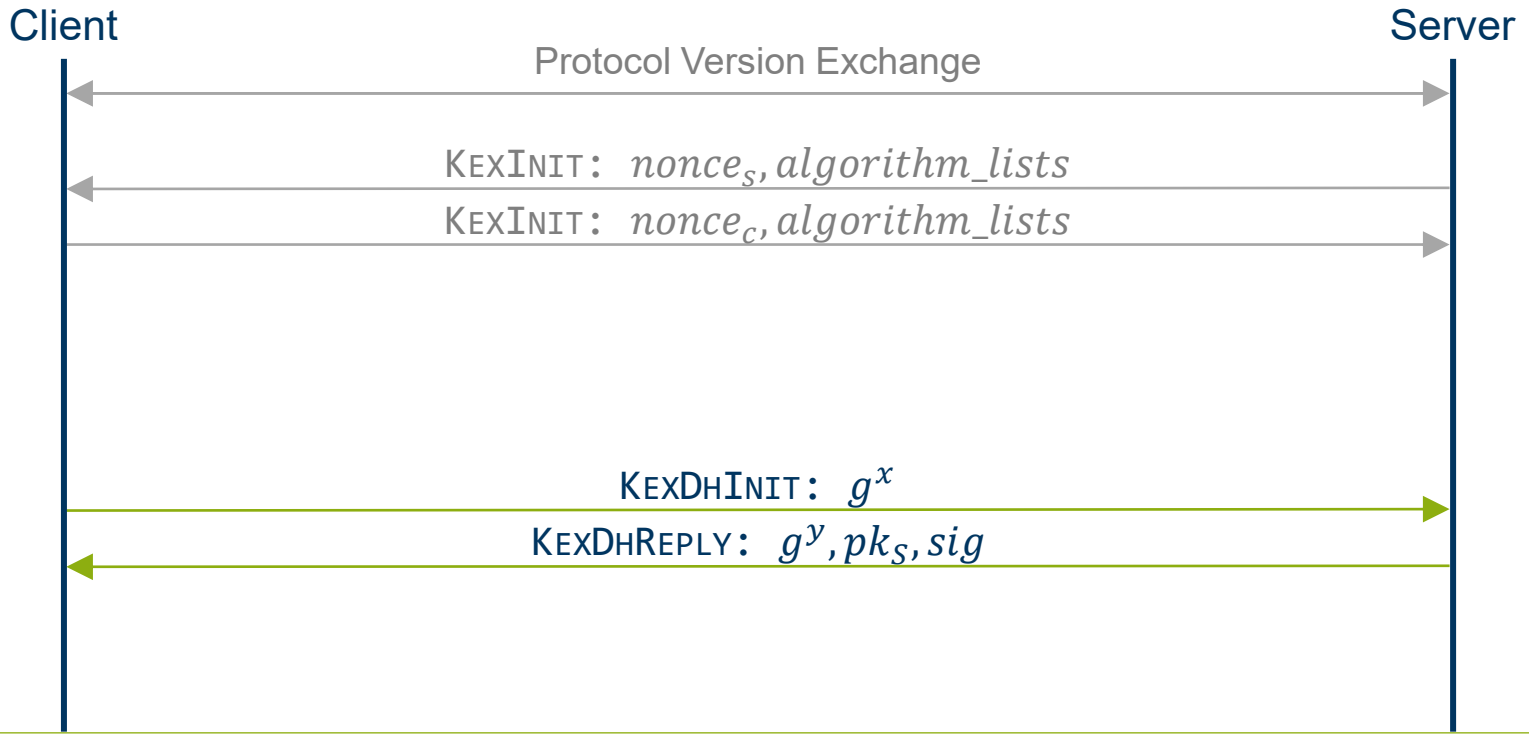
SSH TLP: Protocol Version Exchange



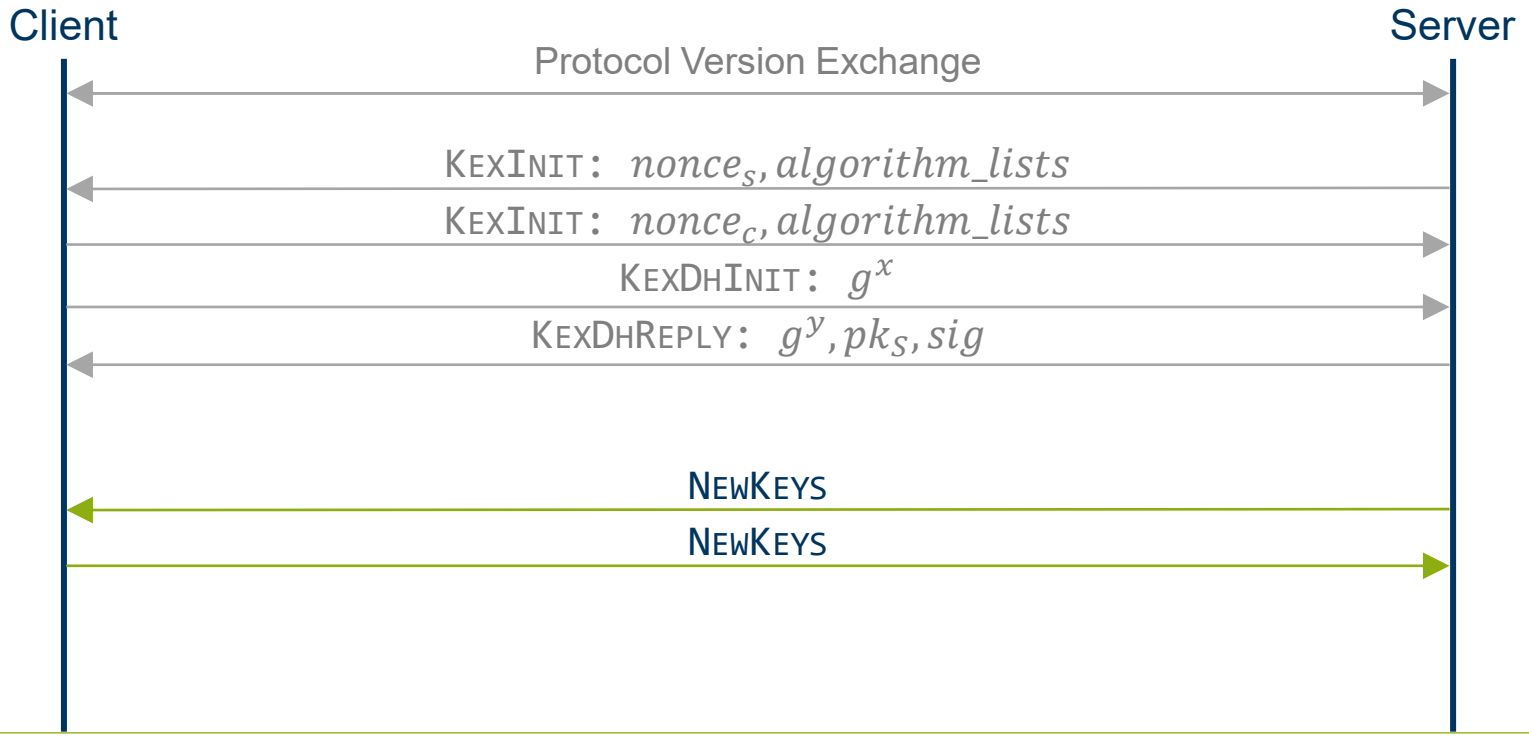
SSH TLP: Algorithm Negotiation



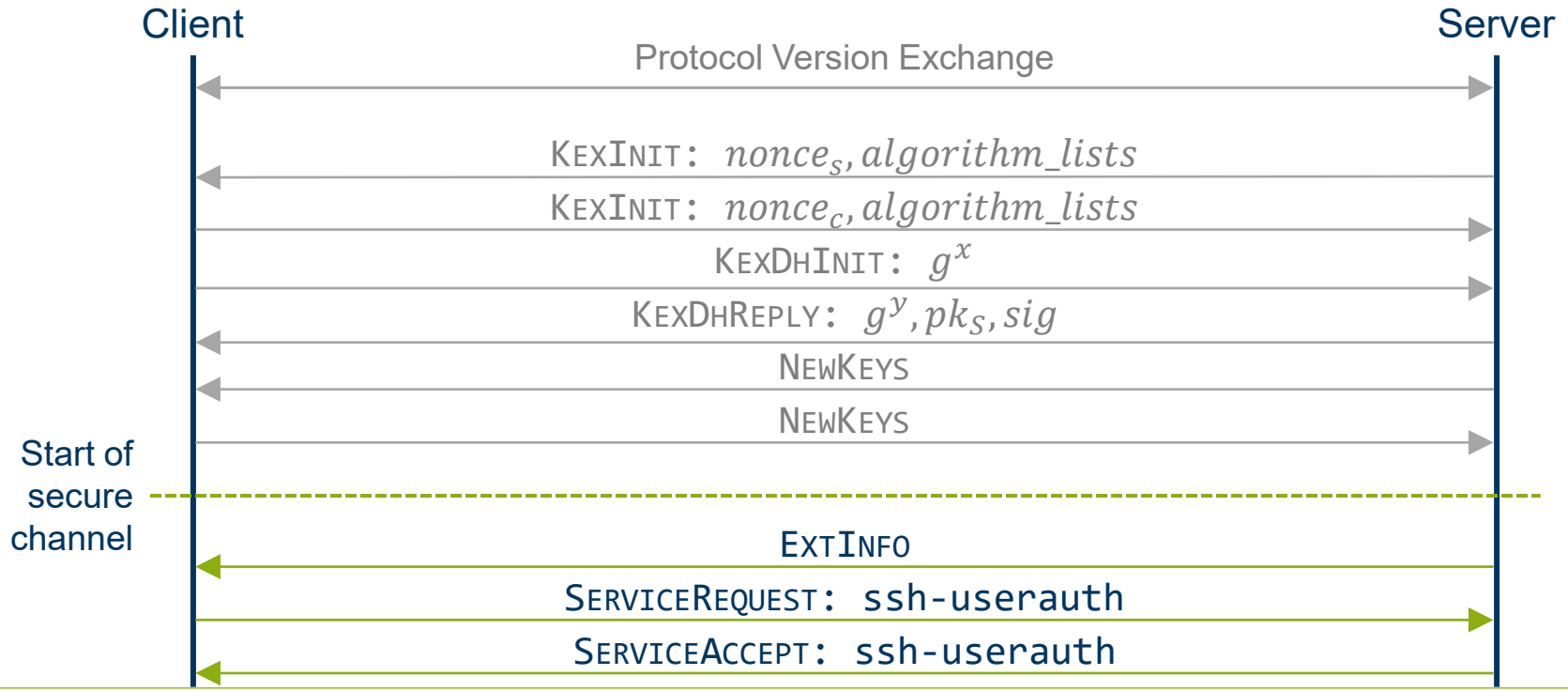
SSH TLP: (DH) Key Exchange



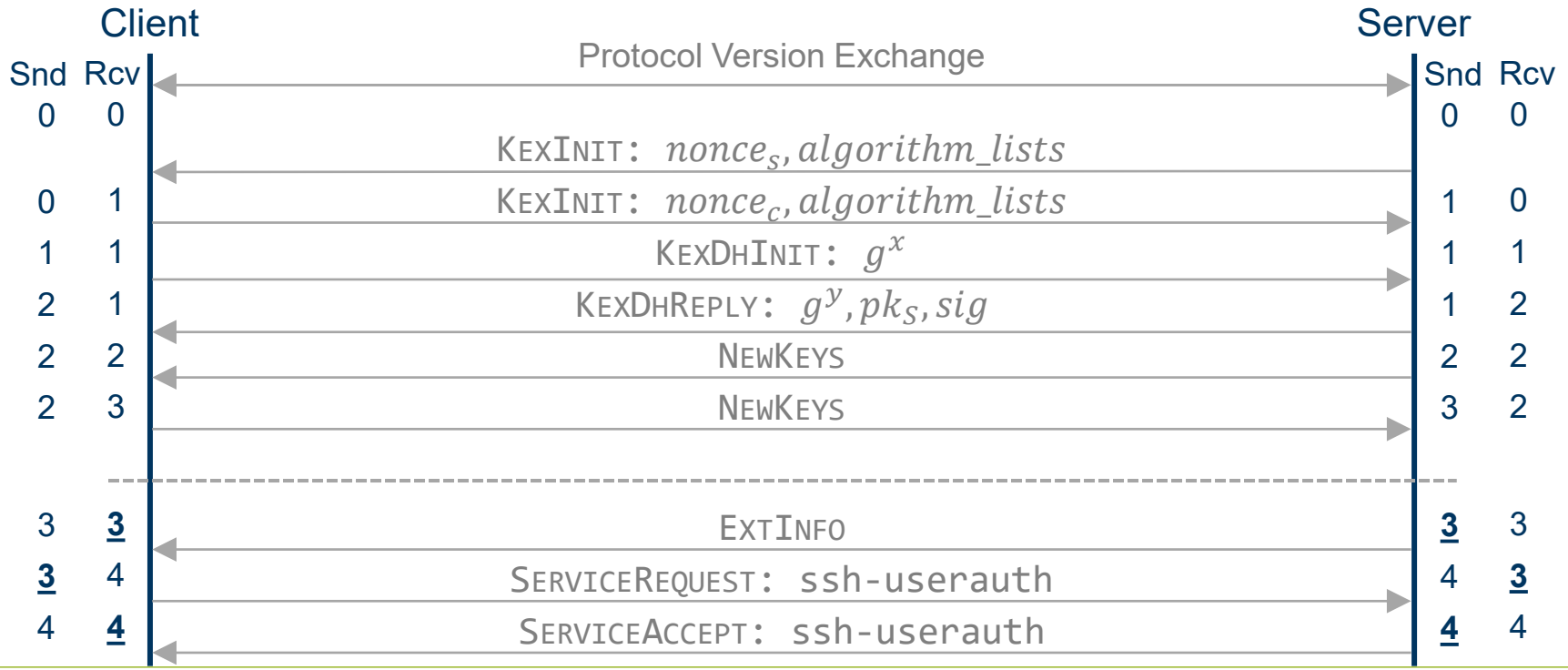
SSH TLP: Activating the Secure Channel



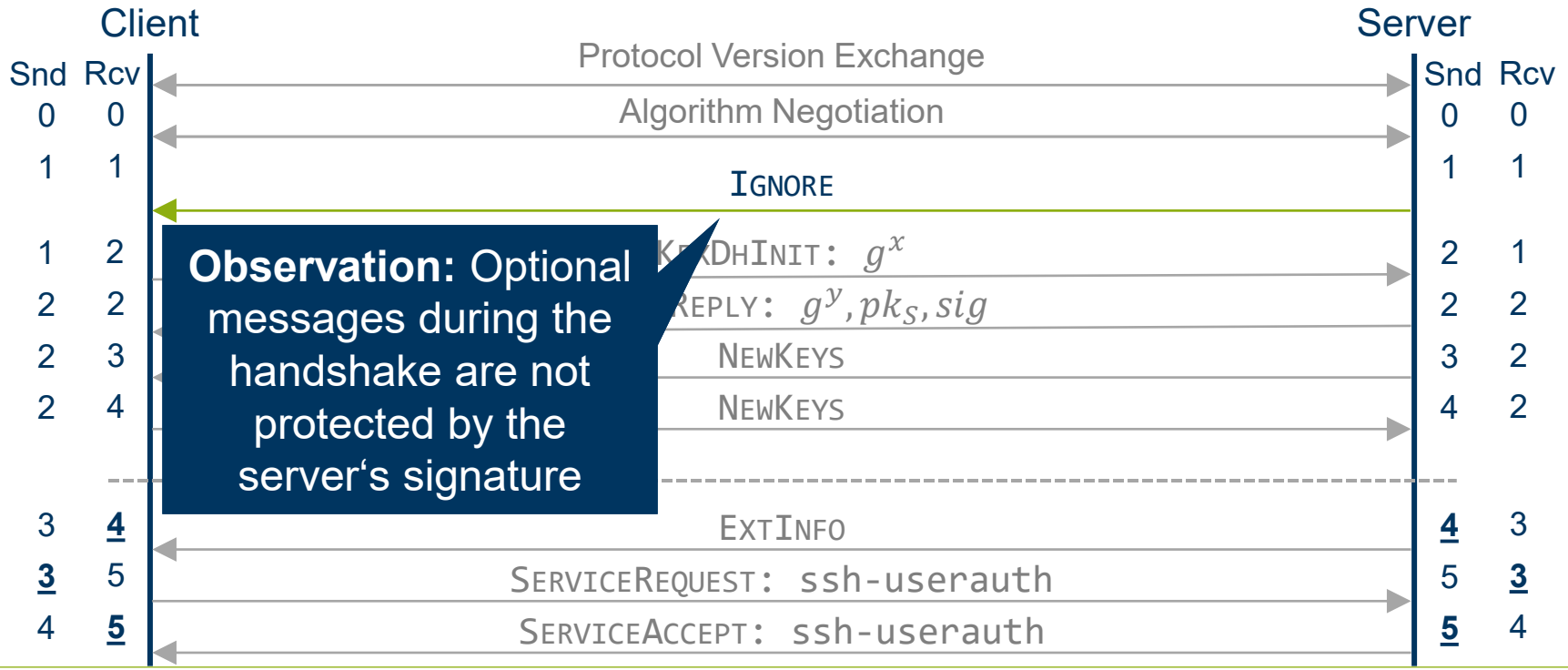
SSH TLP: Requesting Another Service



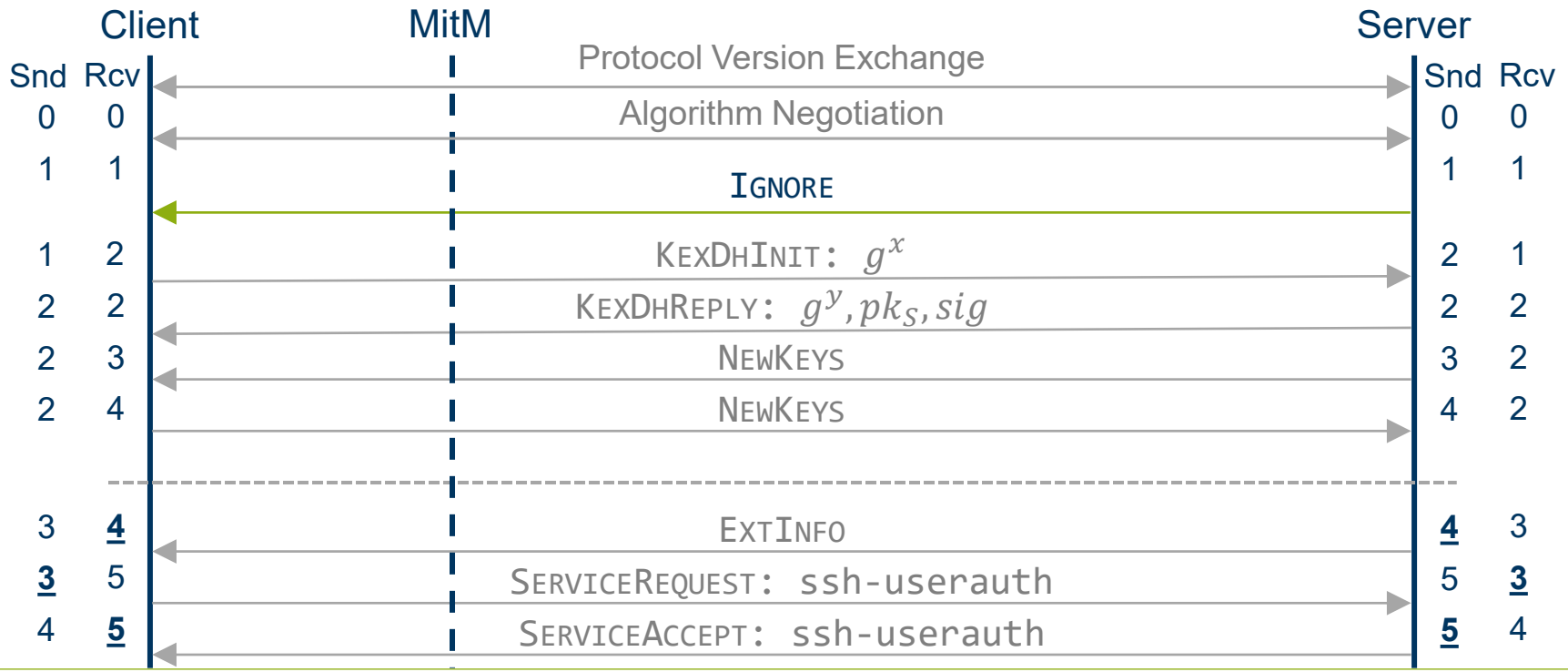
SSH Uses Implicit Sequence Numbers



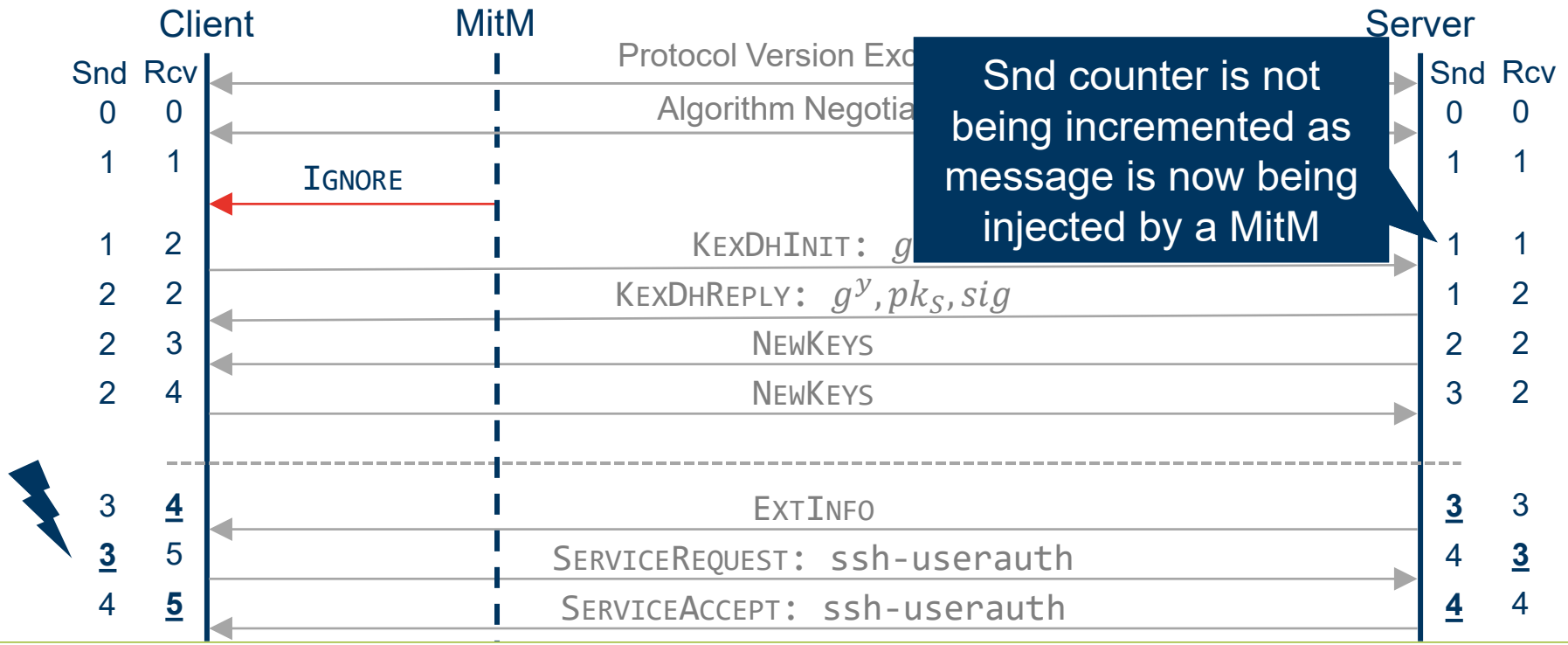
SSH Allows for Optional Messages in Handshakes



SSH Allows for Optional Messages in Handshakes

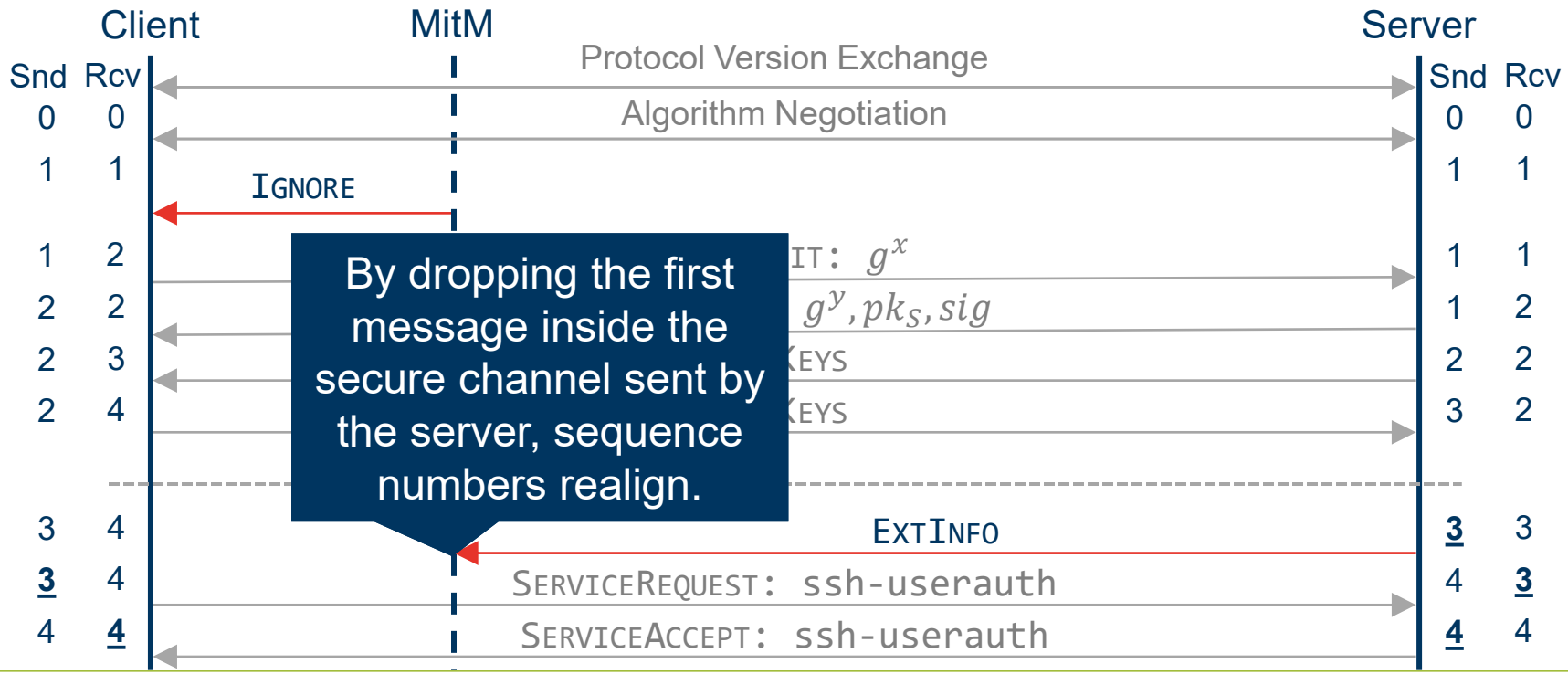


MitM Attackers Can Inject Messages Into Handshake…



… And Drop Messages Inside The Secure Channel

CVE-2023-48795
(CVE-2024-41909)

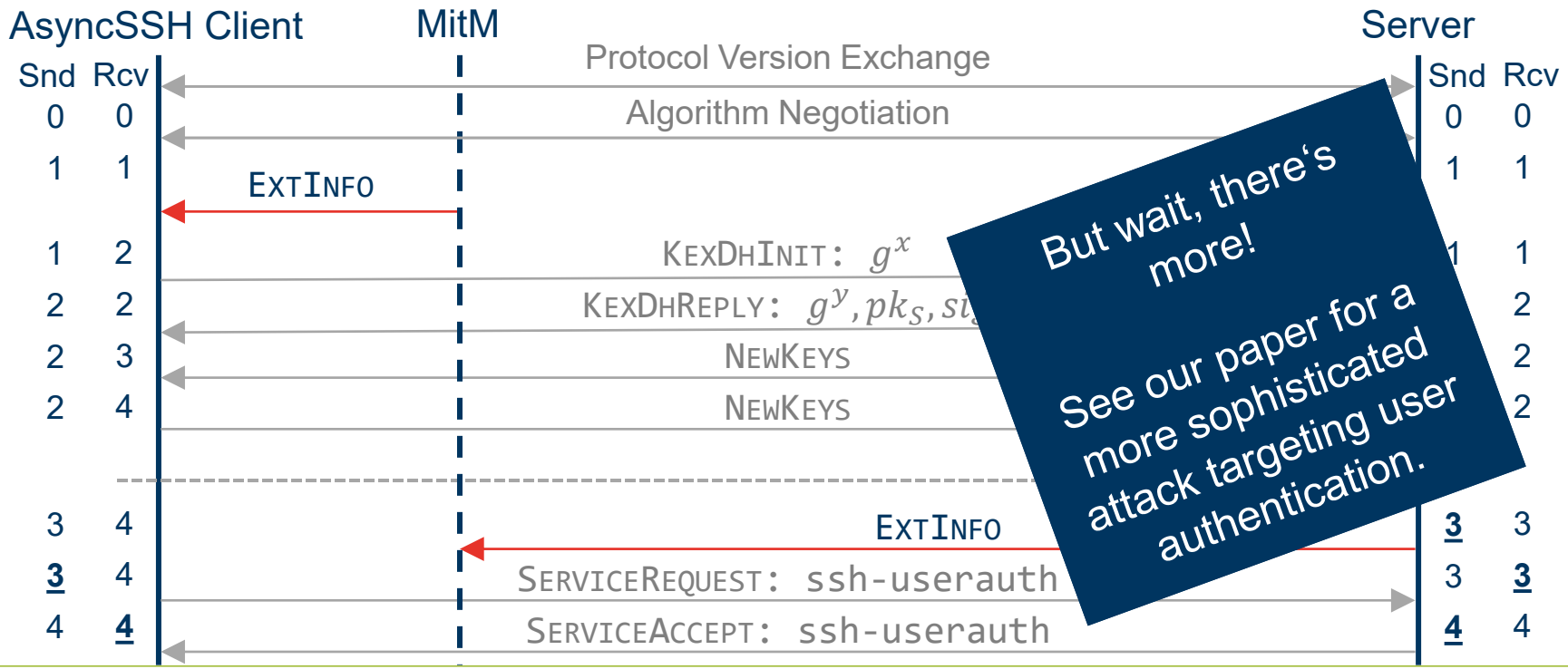


The EXTINFO Message Contains Extensions as Key-Value Pairs

server-sig-algs	ping@openssh.com	Other Extensions
• List of public key algorithms for user authentication • Enables RSA-SHA2 support	• Like Heartbeat extension in TLS • Can be used to obscure keystroke timings	• Not considered because no security impact

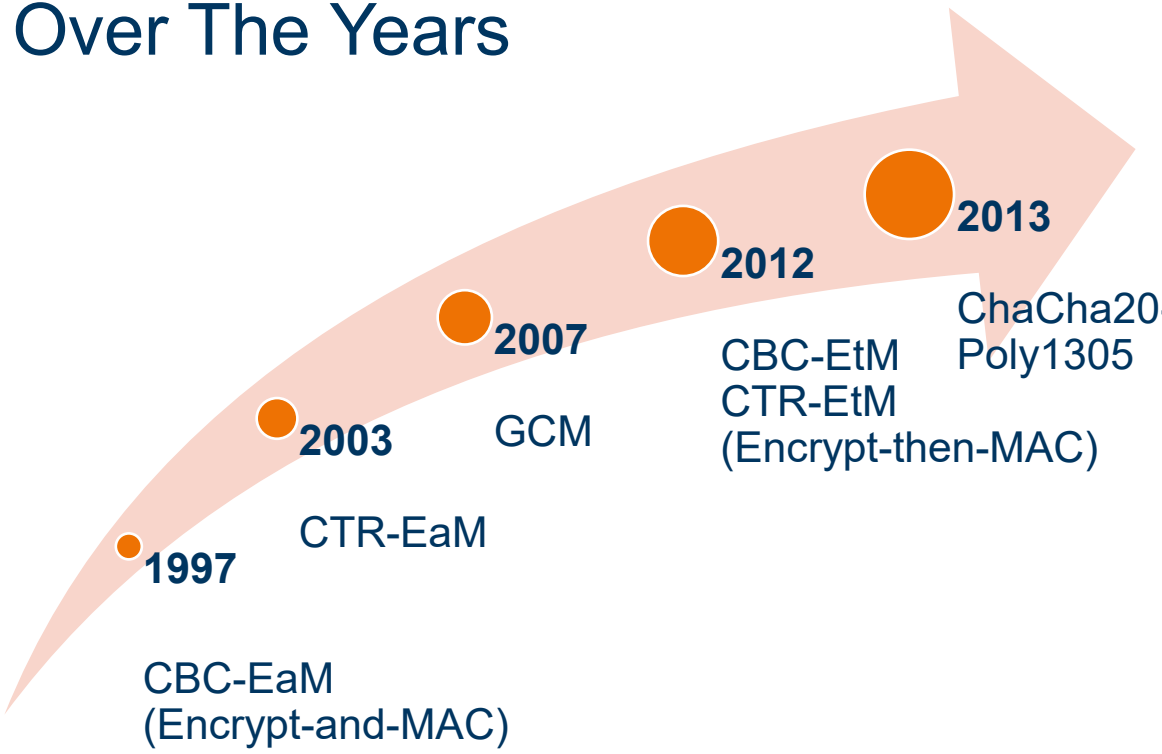
Implementation Bugs Can Escalate Impact

CVE-2023-46445
CVE-2023-46446

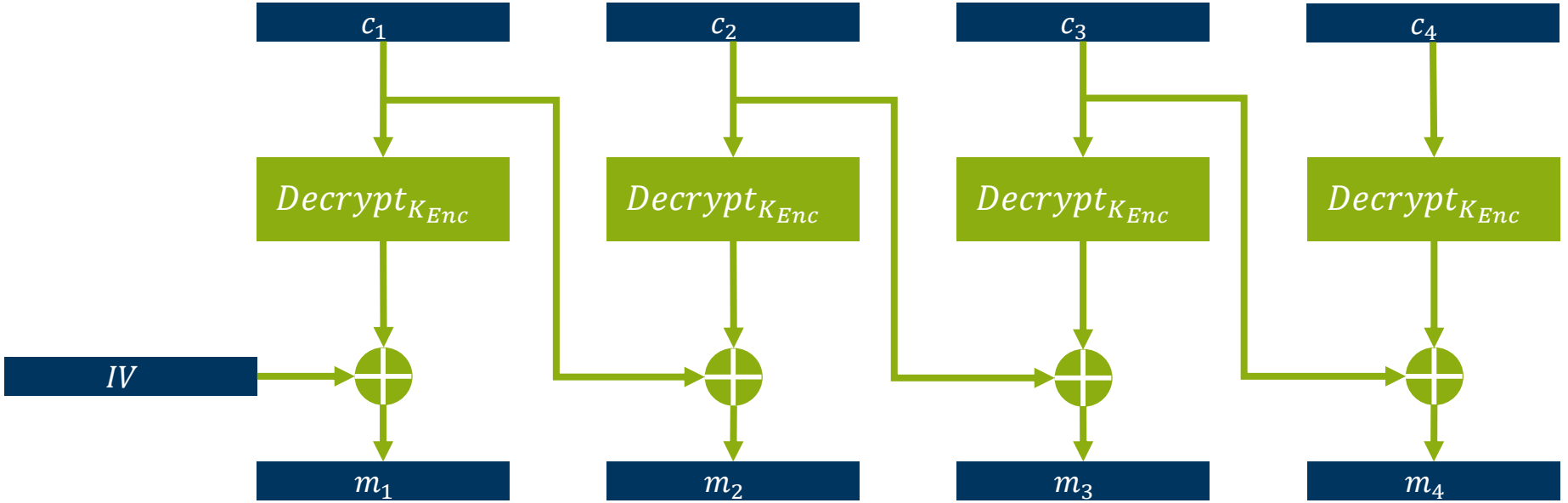


RUHR UNIVERSITÄT BOCHUM RUB

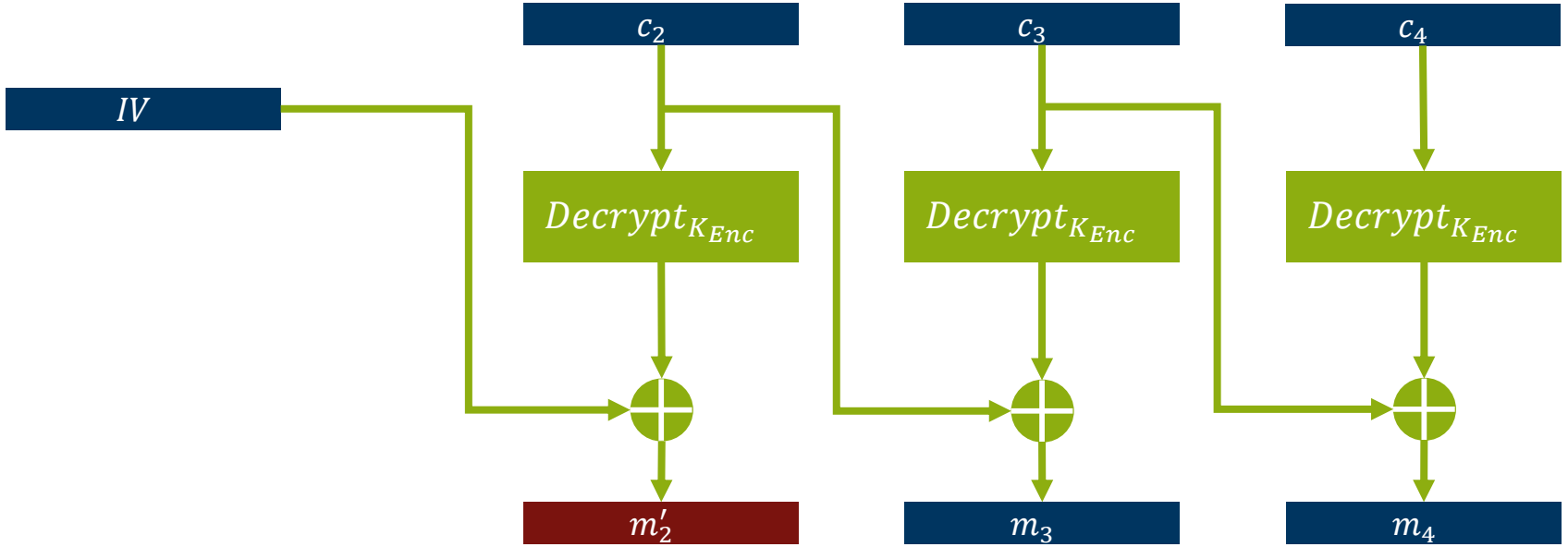
SSH Adopted Various Authenticated Encryption Modes Over The Years



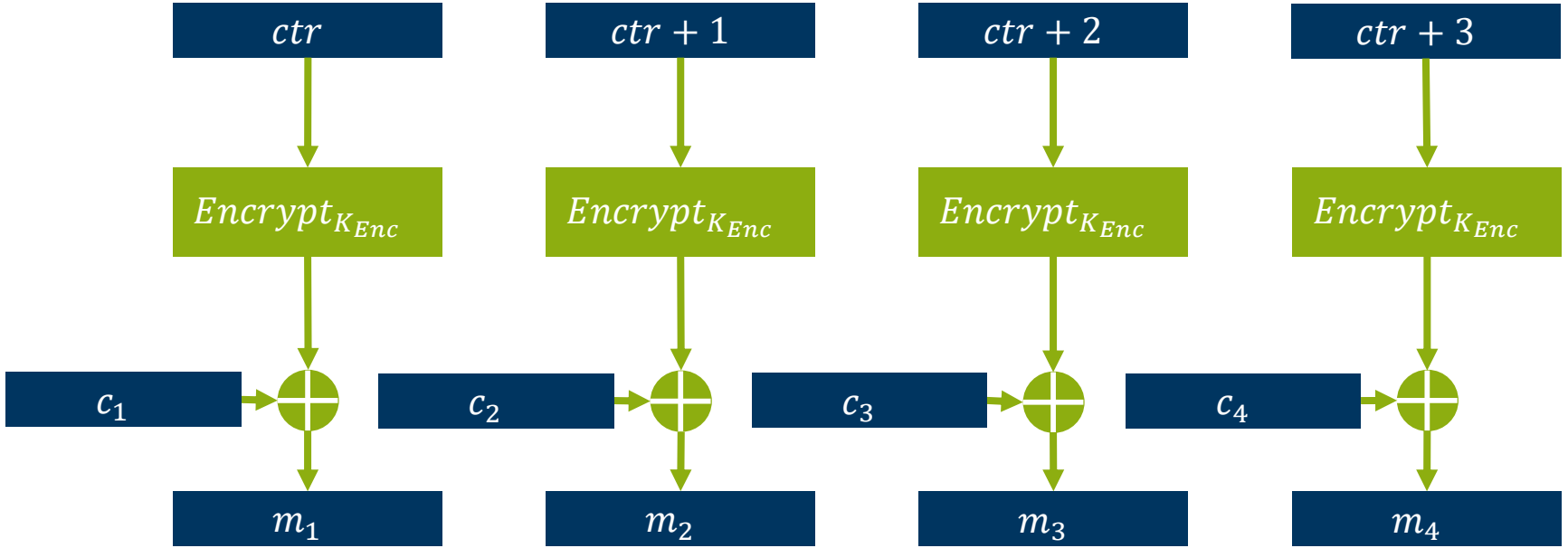
Truncation in CBC Encryption Modes Cause One Pseudorandom Block



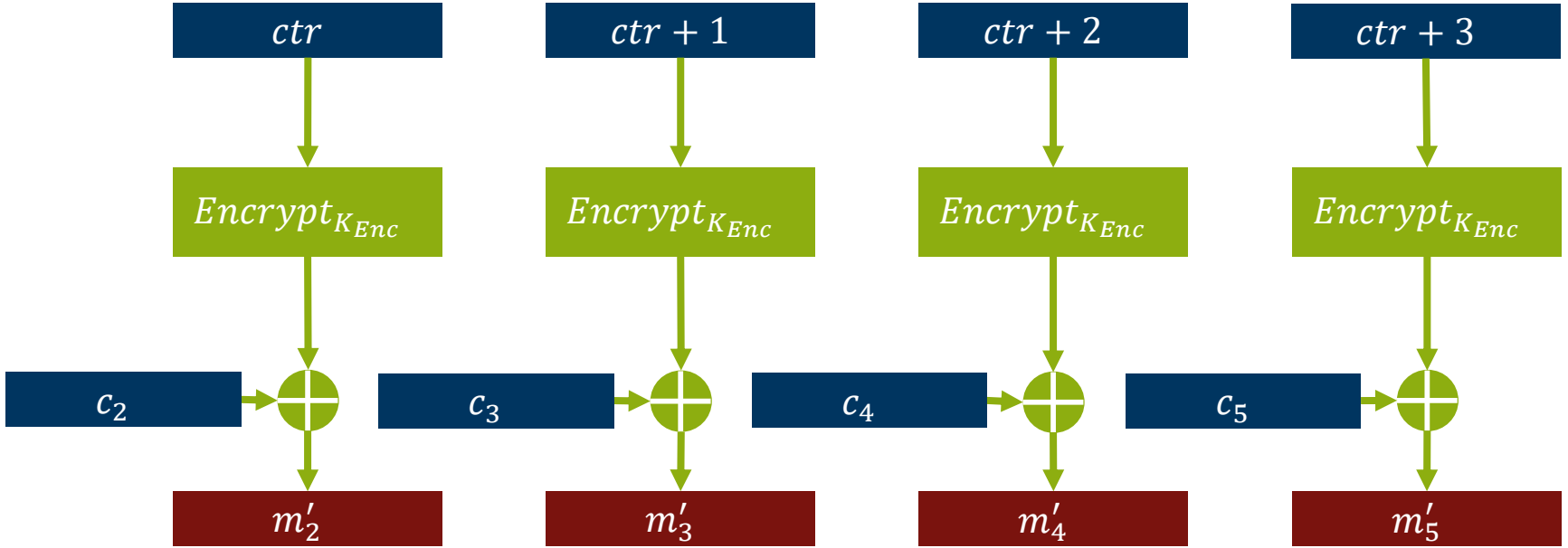
Truncation in CBC Encryption Modes Cause One Pseudorandom Block



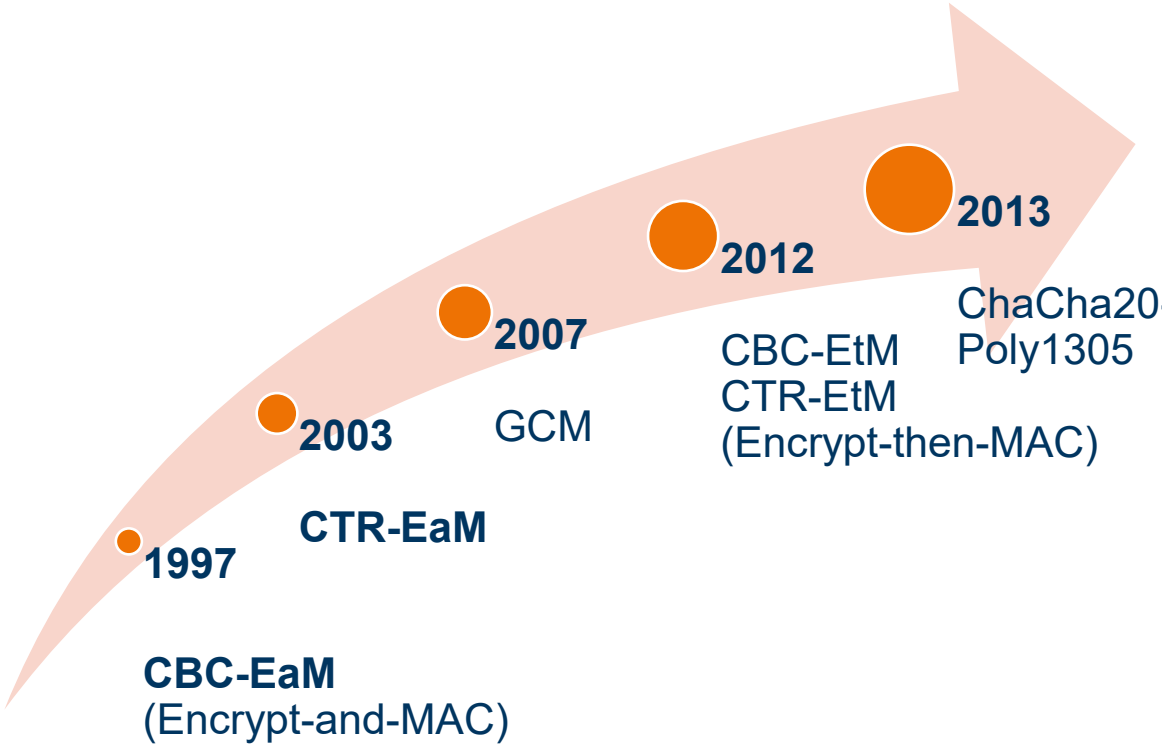
Truncation in CTR Encryption Modes Cause Subsequent Blocks To Become Pseudorandom



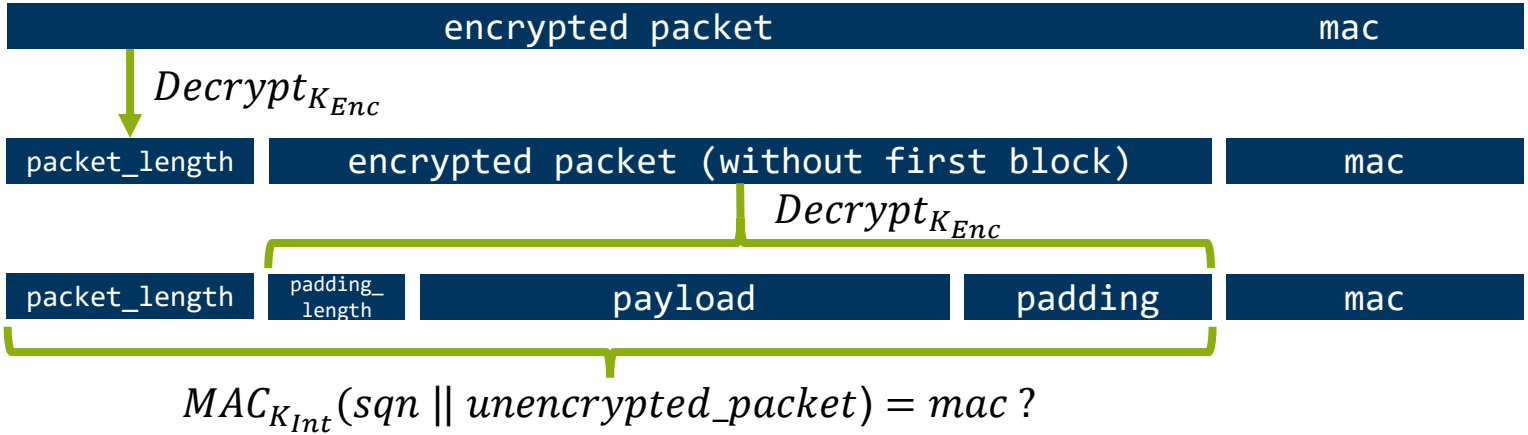
Truncation in CTR Encryption Modes Cause Subsequent Blocks To Become Pseudorandom



Analysis of Encryption Modes – CBC/CTR-EaM

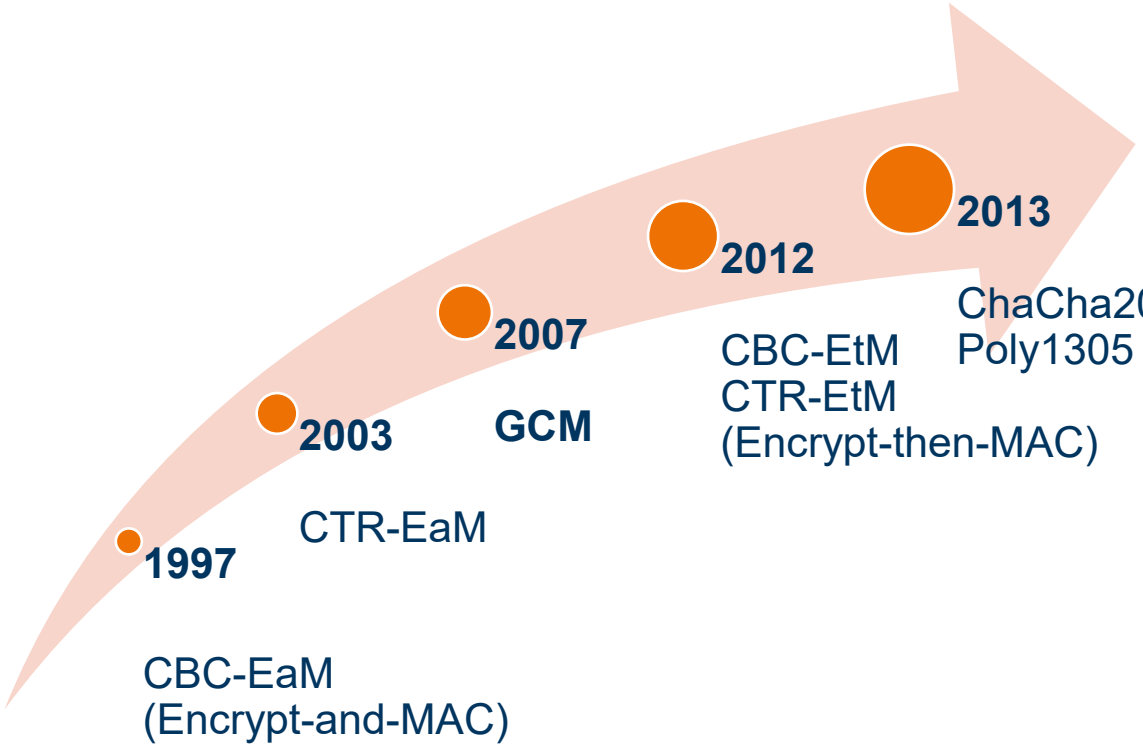


CBC/CTR-EaM Is Not Affected By Our Attacks

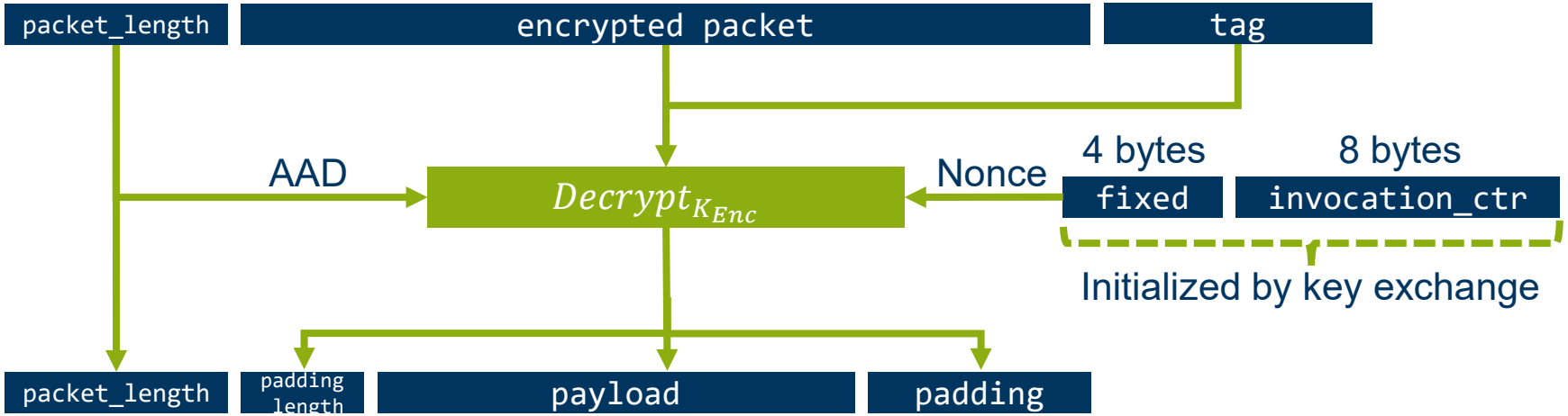


- **Observation:** Truncation of first message causes (at least) the first block of second message to become pseudorandom
- MAC protects integrity of plaintext causing MAC verification failure on truncation

Analysis of Encryption Modes – GCM

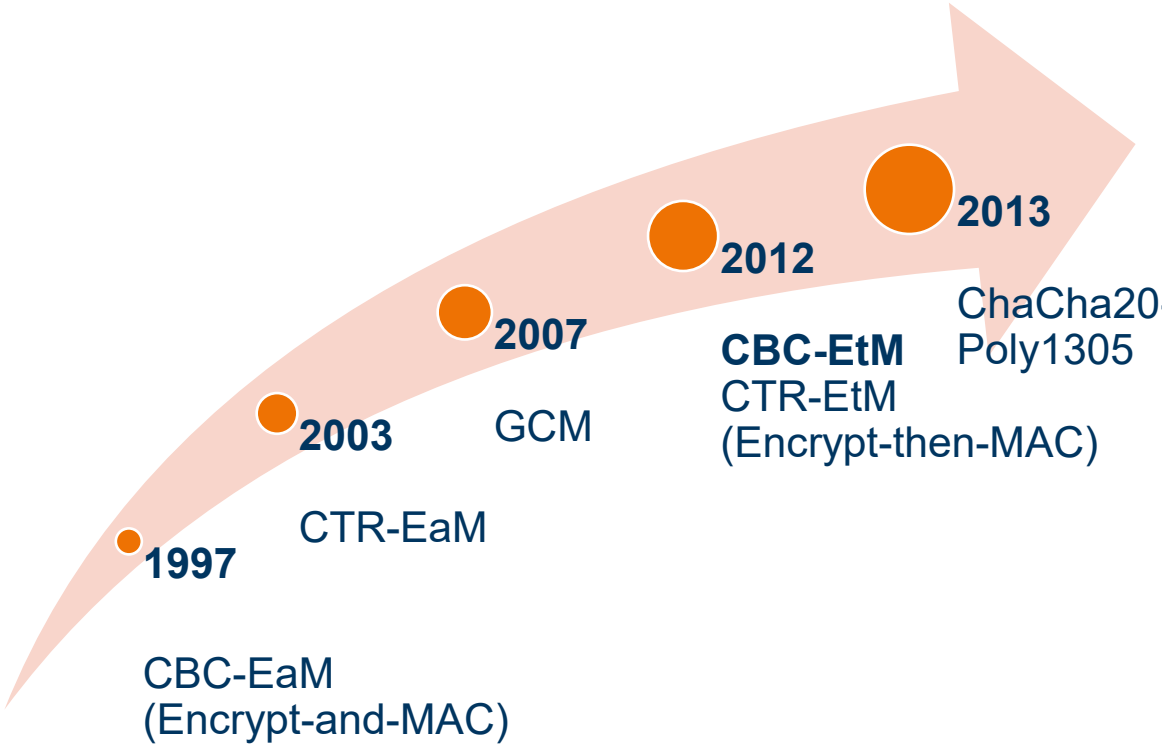


AES-GCM Does Not Use Sequence Numbers

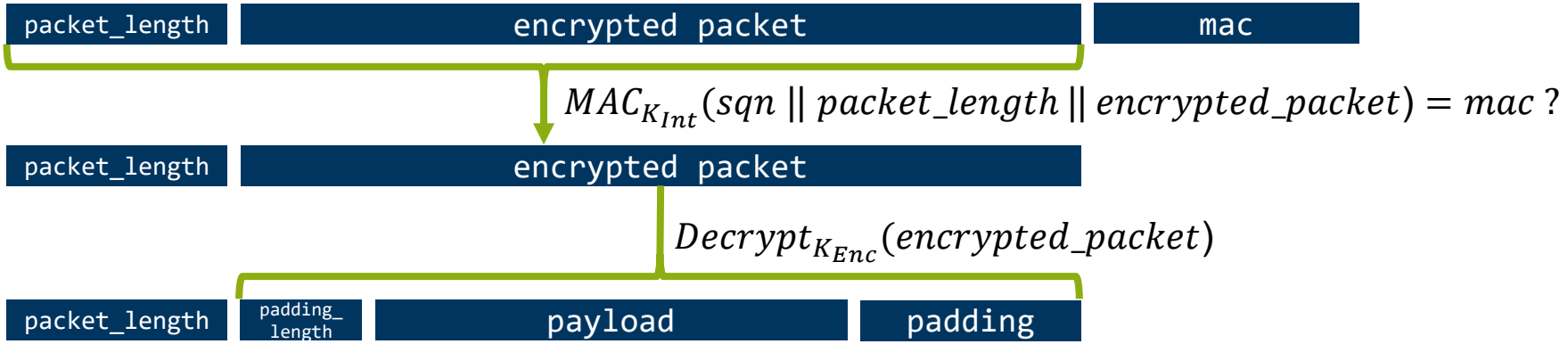


- **Observation:** AES-GCM does not use sequence number but an invocation counter securely initialized through key derivation

Analysis of Encryption Modes – CBC-EtM

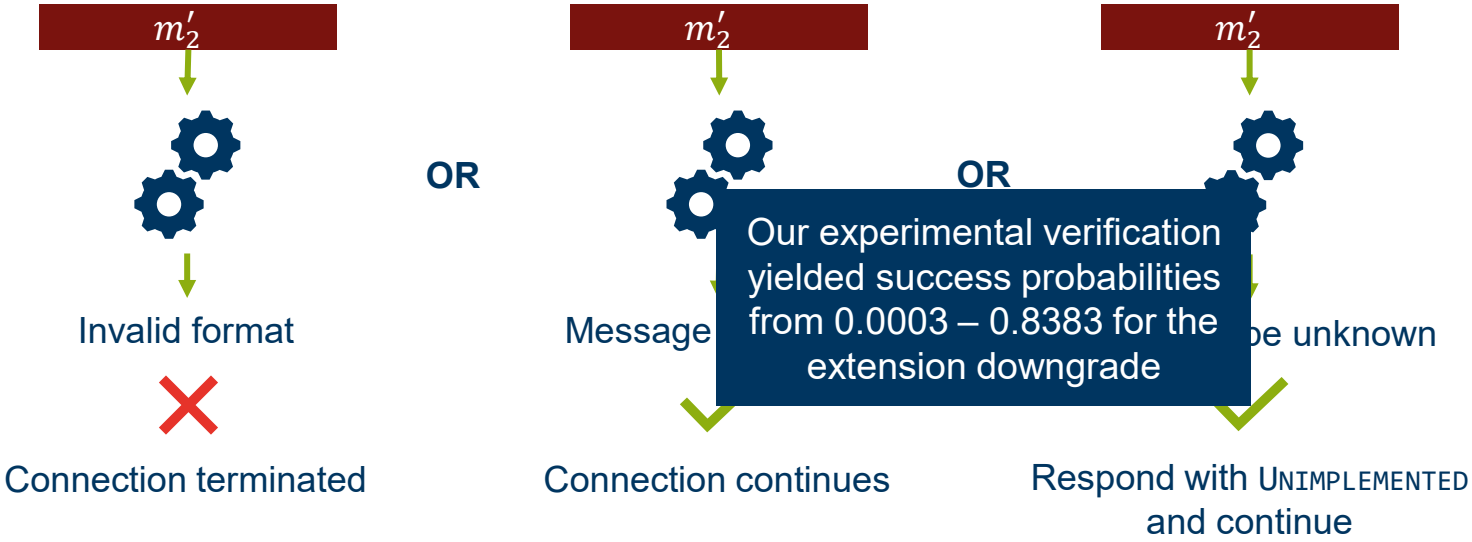


CBC-EtM Allows Probabilistic Truncation Attacks

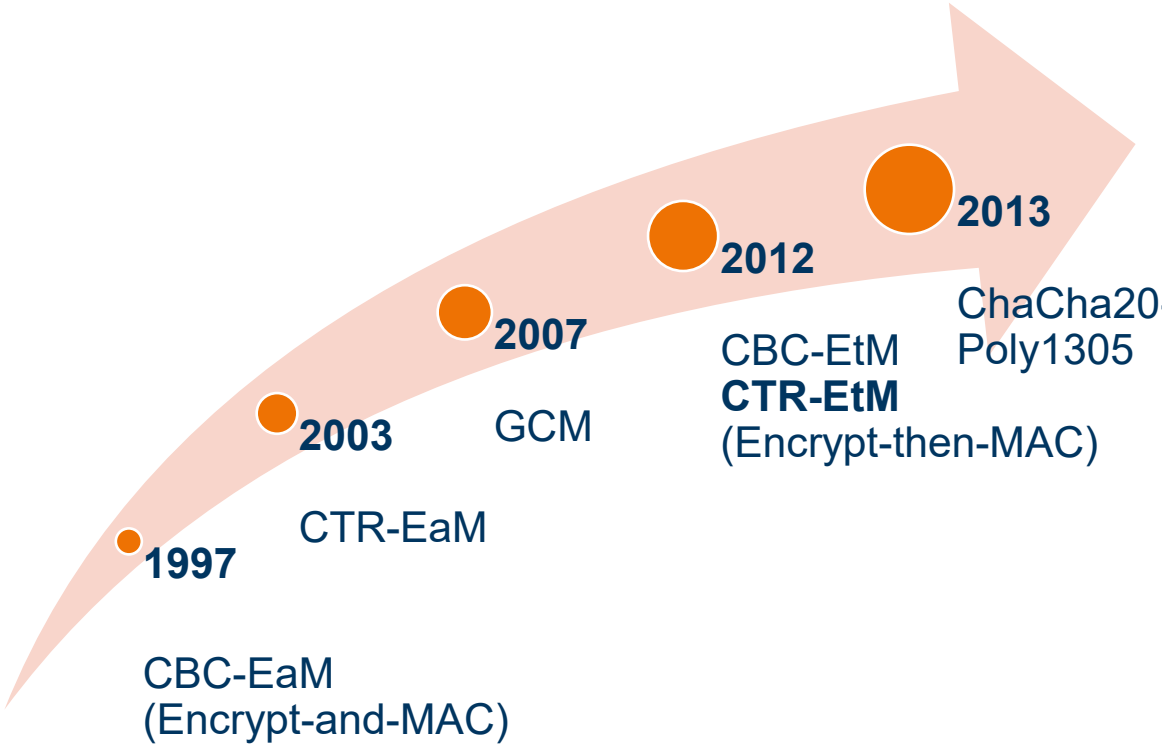


- **Observation:** Truncation of first message causes first block of second message to become pseudorandom
- MAC protects integrity of ciphertext allowing MAC verification to succeed

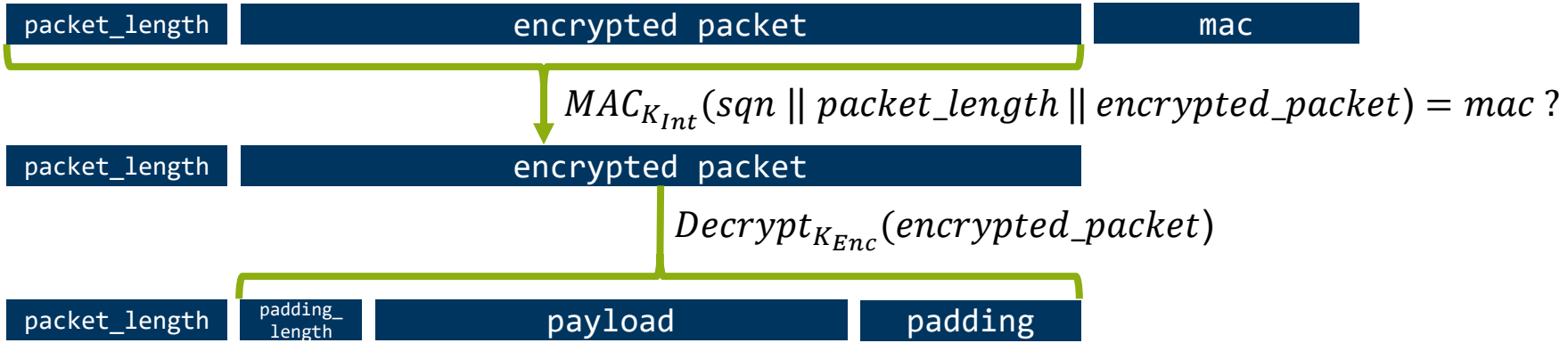
The Attack's Success Depends on How Peers Handle The Corrupt Message Block



Analysis of Encryption Modes – CTR-EtM

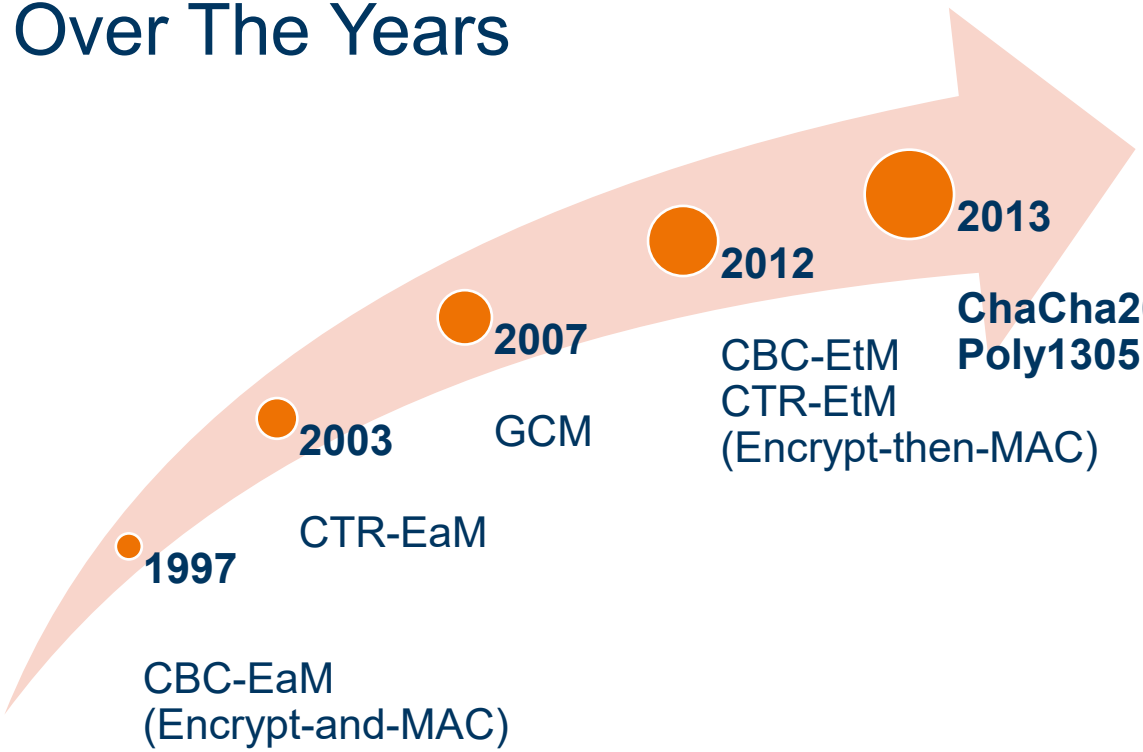


Practical Prefix Truncation with CTR-EtM Is Unlikely



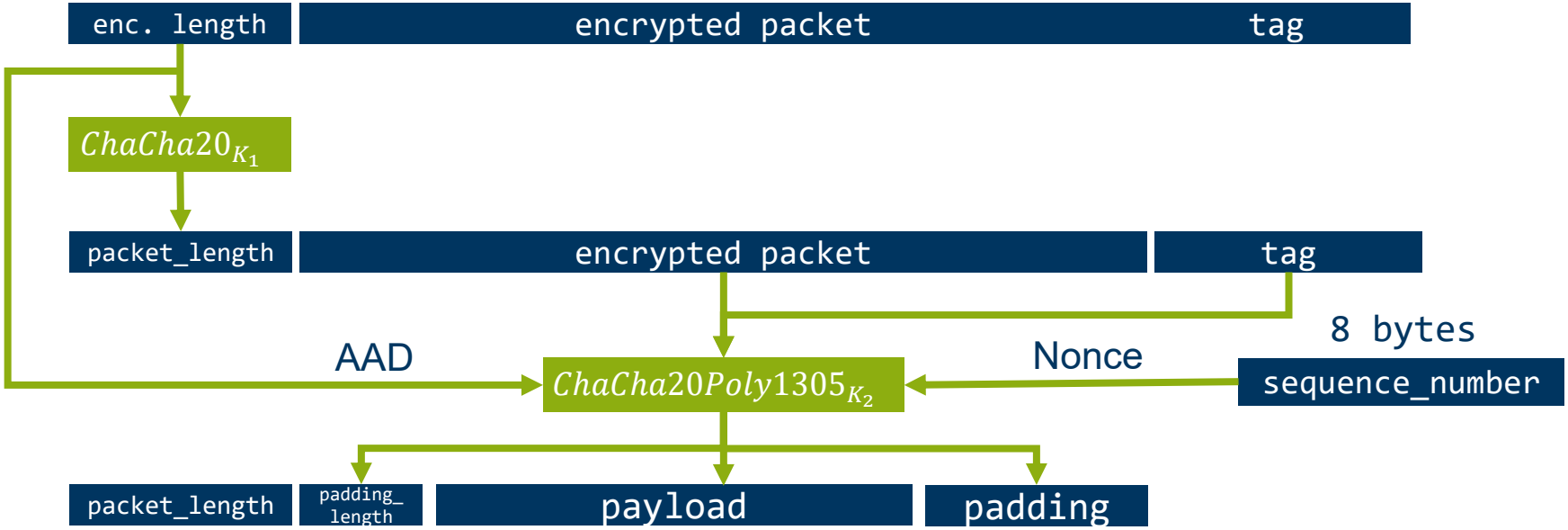
- **Observation:** Truncation of first message causes subsequent blocks to become pseudorandom due to desynchronized keystream
- MAC verification succeeds with same rational as for CBC-EtM
- Connection will eventually terminate on the application layer

SSH Adopted Various Authenticated Encryption Modes Over The Years



ChaCha20-Poly1305 Allows Perfect Prefix Truncation

$$K_{Enc} = K_2 || K_1$$



Successful Prefix Truncation Depends on Authenticated Encryption Mode

Authenticated Encryption Mode		Enc. State	Dec. State	Affected	Exploitable
Encrypt-and-MAC	CBC	$(IV, \mathbf{Snd})$	$(IV, \mathbf{Rcv})$	✗	○
	CTR	$(ctr, \mathbf{Snd})$	$(ctr, \mathbf{Rcv})$	✗	○
Encrypt-then-MAC	CBC	$(IV, \mathbf{Snd})$	$(IV, \mathbf{Rcv})$	✓	◐
	CTR	$(ctr, \mathbf{Snd})$	$(ctr, \mathbf{Rcv})$	✓	◐
GCM		$ctr_{Invocation}$	$ctr_{Invocation}$	✗	○
ChaCha20-Poly1305		**Snd**	**Rcv**	✓	●

ChaCha20-Poly1305 And EtM Are Popular

AE Mode	Preferred		Supported	
ChaCha20-Poly1305	8,739k	57.64%	10,247k	67.58%
CTR-EaM	3,964k	26.14%	4,200k	27.70%
GCM	1,219k	8.04%	10,450k	68.92%
CTR-EtM	828k	5.46%	10,685k	70.46%
CBC-EaM	359k	2.37%	1,585k	10.46%
CBC-EtM	14k	0.09%	2,614k	17.24%
Other	2k	0.01%	-	-
Unknown / No KEXINIT	36k	0.24%	-	-
Total	15,164k	100%		

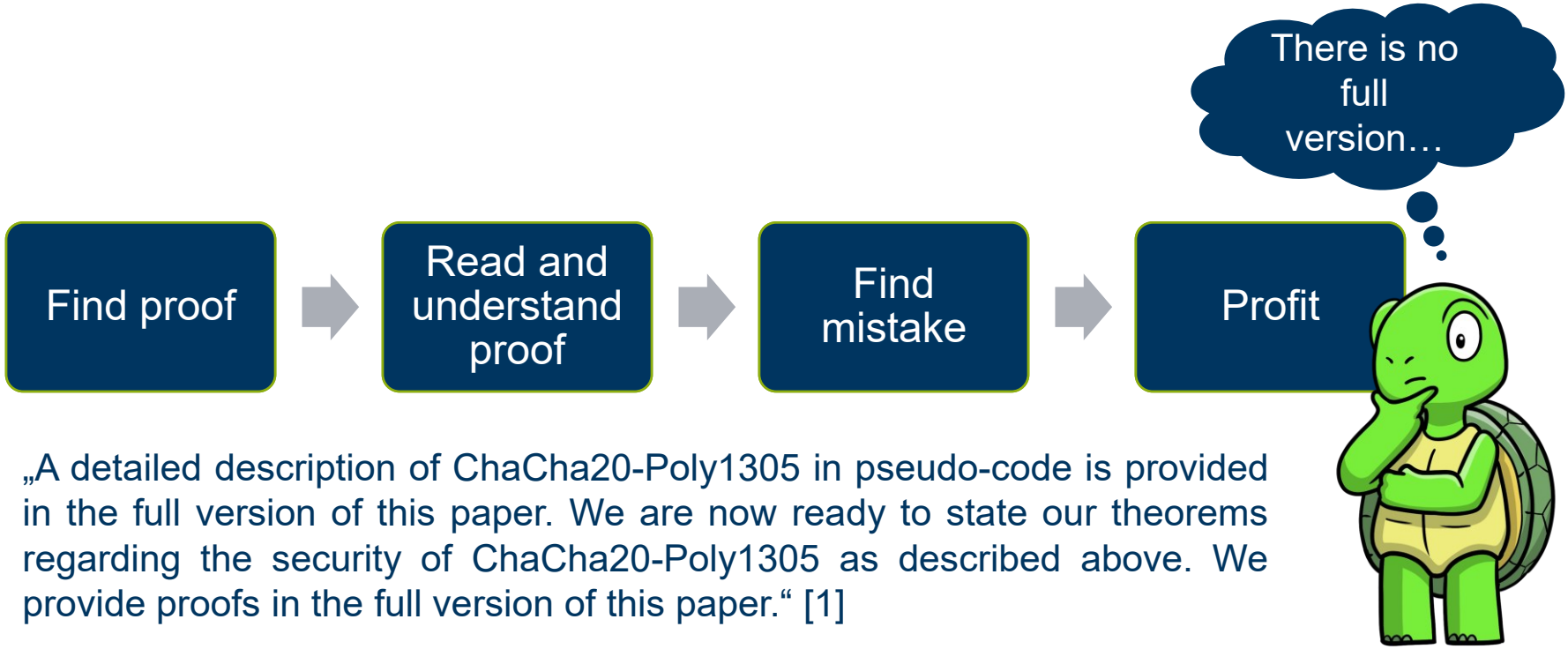
SSH TLP Has Been Proven Secure

- **Security of the handshake**
  - Williams (IMACC 2011): SSH handshake with DH key exchange
  - Bergsma et al. (CCS 2014): Multi-ciphersuite security
- **Security of the secure channel**
  - Bellare et al. (CCS 2002): Encrypt-and-MAC
  - Paterson, Watson (EUROCRYPT 2010): Encrypt-and-MAC with CTR-Mode
  - Albrecht et al. (CCS 2016): **Encrypt-then-MAC**, AES-GCM, **ChaCha20-Poly1305**

Analyzing Security Proofs May Not Be Straightforward

Read and understand proof → Find mistake → Profit

Analyzing Security Proofs May Not Be Straightforward



„A detailed description of ChaCha20-Poly1305 in pseudo-code is provided in the full version of this paper. We are now ready to state our theorems regarding the security of ChaCha20-Poly1305 as described above. We provide proofs in the full version of this paper.“ [1]

Analyzing Security Proofs May Not Be Straightforward

Talk to authors → Find proof → Read and understand proof → Find mistake → Profit

- Extensive research and talking to one of the co-authors revealed full proof can be found in the PhD thesis of Hansen [2]

Proof Abstractions Assume Sequence Numbers Are Zero-Initialized [2]

alg. ssh-ChaCha20-Poly1305-Gen

1 : $\mathsf{seqnr} = 0$
2 : $\mathsf{frag} = \epsilon$
3 : $\mathsf{CLOSED} = \mathsf{false}$
4 : $\mathsf{k} \leftarrow\!\$\ \mathsf{B}^{64}$
5 : $\sigma = \mathsf{seqnr}$
6 : $\varrho = (\mathsf{frag}, \mathsf{seqnr}, \ell_{\mathsf{packet}}, \mathsf{CLOSED})$
7 : **return** $(\mathsf{k}, \sigma, \varrho)$

alg. ssh-fgEtM-Gen

1 : $\mathsf{seqnr} = 0$
2 : $\ell_{\mathsf{packet}} = 0$
3 : $\mathsf{frag} = \epsilon$
4 : $\mathsf{CLOSED} = \mathsf{false}$
5 : $\mathsf{k}_e \leftarrow \mathsf{Gen_e}$
6 : $\mathsf{k}_m \leftarrow \mathsf{Gen_m}$
7 : $\mathsf{k} = \mathsf{k}_e \parallel \mathsf{k}_m$
8 : $\sigma = \mathsf{seqnr}$
9 : $\varrho \leftarrow (\mathsf{frag}, \mathsf{seqnr}, \ell_{\mathsf{packet}}, \mathsf{CLOSED})$
10 : **return** $(\mathsf{k}, \sigma, \varrho)$

Mitigating Our Attack Is Difficult

Countermeasure	Our Suggestion	"Strict KEX" (OpenSSH)
Reset sequence numbers at key installation	✓	✓
Authenticate the entire handshake transcript (hash)	✓	
Harden handshake to disallow unexpected messages		✓

> 30 unique implementations support "strict kex"

~ 11 million servers offer "strict kex"

Lessons Learned

- **Terrapin is a novel cryptographic attack targeting SSH channel integrity**
  - Can be exploited in practice to downgrade the connection‘s security
  - May lead to more severe vulnerabilities if combined with state machine flaws

- Affected modes of encryption (% Supported):
  - ChaCha20-Poly1305 (67.58%)
  - CBC-EtM (17.24%)
  - CTR-EtM (70.46%)

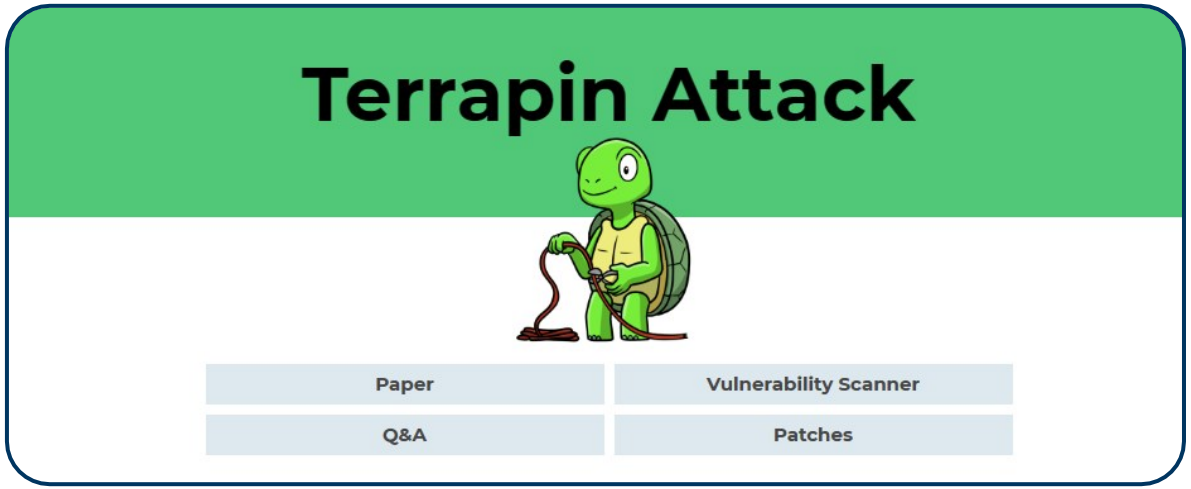
- All these modes have been proven secure in previous works
  - Proofs hold when “strict kex” countermeasure applied

References

[1] Albrecht, M. R., Degabriele, J. P., Hansen, T. B., & Paterson, K. G. (2016, October). A surfeit of SSH cipher suites. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1480-1491).

[2] Hansen, T. B. (2020). *Cryptographic Security of SSH Encryption Schemes* (Doctoral dissertation, Royal Holloway, University of London).

Thanks! Questions?



https://terrapin-attack.com/

E-Mail: fabian.baeumer@rub.de
X (formerly Twitter): @TrueSkrillor
Mastodon: @Skrillor@infosec.exchange