Kleptography in Authentication Protocols: Why is it Still Possible?

Carina Heßeling
carina.hesseling@fernuni-hagen.de
FernUniversität in Hagen
Hagen, Germany

Jörg Keller
joerg.keller@fernuni-hagen.de
FernUniversität in Hagen
Hagen, Germany

Sebastian Litzinger
sebastian.litzinger@fernuni-hagen.de
FernUniversität in Hagen
Hagen, Germany

ABSTRACT

Network authentication frequently relies on nonces, and even widely deployed protocols still rely on random nonces, although they might enable kleptography attacks. Notably, for TLS a kleptography-based covert channel has been published, and despite a proposal to cure this weakness via controlled randomness including backward compatibility, the protocol description is not updated. We investigate if lack of bandwidth, i.e., lack of applicability, could be a reason not to care for such an update. Moreover, we give examples of other authentication protocols that might suffer from a similar weakness, and that possibly might profit from a similar cure, thus indicating necessity of further research.

CCS CONCEPTS

• **Security and privacy → Authentication**; *Pseudonymity, anonymity and untraceability*; • **Theory of computation → Cryptographic primitives**; • **Social and professional topics → Computer crime**.

KEYWORDS

network authentication, kleptography attack, covert channel, network steganography

ACM Reference Format:
Carina Heßeling, Jörg Keller, and Sebastian Litzinger. 2022. Kleptography in Authentication Protocols: Why is it Still Possible?. In *EICC '22: European Interdisciplinary Cybersecurity Conference, June 15–16, 2022, Barcelona, Spain.* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

"Classic" texts on networks and security frequently mention the use of nonces for different reasons such as avoidance of replay attacks [6]. Also, frequently a random choice of a nonce is mentioned, as this is unpredictable for an attacker. However, the use of uncontrolled randomness opens the possibility for data leakage via a kleptography attack [11] that enables a network covert channel [5], usually for criminal purpose. Put briefly, in a kleptography attack other communication partners cannot check if a randomly looking number has indeed been generated as a random number, or if, e.g., a compromised library (seen as a black box) incorporated encrypted information only usable for an attacker that knows how to interpret it (e.g., who has a key to decrypt it). Also, measures have been published to cure this problem by "controlled" randomness that can be verified by communication partners [1, 2]. In particular, TLS as a prominent protocol has authentication defined with random nonces, a covert channel exploiting these nonces has been published (cf. [3] and the references therein), and a proposal how to cure the protocol (including backward compatibility) has been given as well [2]. Still, the actual protocol specification [8] does not reflect this, as the *ClientHello* message in the key exchange phase includes a random nonce.

Reasons why the protocol specification is not updated could be[1]:

(1) The affected protocol is not widely deployed.
(2) It is unclear how to exploit the weakness.
(3) It is unclear how to cure the protocol from the weakness.
(4) Exploitation does not bring real advantage as bandwidth is too small.

The reasons are ordered such that if reason (i) is true, then we can stop wondering, otherwise we must check the next reason. If none of the reasons is true, we have no proof that the list is complete, but could not find another reason. Given the above story, we believe that the first three reasons can be excluded, and thus concentrate on reason (4).

After briefly summarizing background information in Section 2, we will argue in Section 3 that the last reason also can be excluded. We conclude and give an outlook to future research in Section 4.

2 BACKGROUND

Steganography is the art of hiding the existence of some information, in contrast to cryptography where the content of that information is protected from unwanted reading [5]. In network steganography, we consider a situation where two communication partners, which we denote as client and server, communicate via a communication network. In our context, they are also called overt sender and receiver, and their communication is called overt communication or carrier.

Two further entities, the covert sender (CS) and covert receiver (CR), are present in the network. CS sends a secret message to CR, where secret not only refers to the message content but also to the fact that the message has been sent at all. To this end, the secret message is hidden within an innocent overt communication. For example, an unused header bit in a network packet can be used to transfer a bit of the secret message. CS and CR thus establish a network covert channel[2] as a policy-breaking communication that was unforeseen in the original communication system [4].

[1] We exclude reasons like politics, lobbyism, etc.
[2] More exactly, it is a storage channel, we will ignore timing and hybrid channels for brevity.

EICC '22, June 15–16, 2022, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

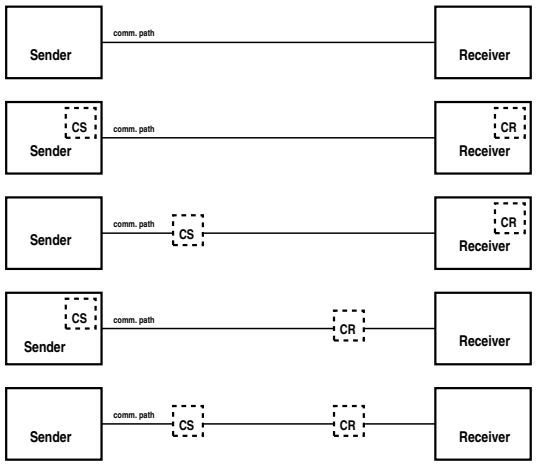

Figure 1: Possible positions of covert sender and receiver on the communication path between overt sender and receiver.

Kleptography [11] means that a value from a black box cryptographic device or software is assumed to be random, but in reality it contains encrypted content when the device is compromised and serves as CS. As encryption turns legible text into randomly looking bitstrings, an observer cannot distinguish the two cases, while CR can decrypt the content. Typically, this type of attack is used for information leakage, i.e., for criminal purpose [5], although [10] use a similar trick to circumvent censorship. Janovsky et al. implemented a kleptography attack [3] for the TLS handshake, where a random nonce is used even in the current version TLS 1.3 [8].

3 COVERT CHANNEL BANDWIDTH ESTIMATION

We can distinguish different possibilities where CS and CR can be positioned, cf. Figure 1. In a scenario for TLS challenge-response authentication, CS is with the sender / challenger. CR might be placed with the receiver / verifier, or on the communication path in between. In the former case, CR only sees challenges directed to itself. In the latter case, CR might see all challenges that CS sends out. Thus, the bandwidth depends on the scenario.

In the case of TLS handshake, the nonce is part of the ClientHello message. Such message is a mandatory part to establish a new connection and of connection resumption [8]. Hence, even in the former case, the challenger/CS can frequently (e.g., once an hour, in order not to look too suspicious) ask for connection resumption by sending a new ClientHello message. In the latter case, the challenger/CS can connect to multiple servers during a day, and CR would see all ClientHello messages. The nonce in a ClientHello message comprises 32 bytes [8]. Given an hourly resumption and 10 different servers to connect to, the bandwidth of such channel will be 6,144 and 61,440 bit/day in the former and latter cases, respectively. Thus, the bandwidth is non-negligible.

4 CONCLUSIONS

We have demonstrated that bandwidths achievable in a covert channel resulting from a kleptography attack against TLS are large enough to make such a channel interesting, thus excluding the last possible reason not to update the TLS specification and heal it from kleptography attack.

TLS might not be the only popular protocol that suffers from this weakness. While a systematic investigation of authentication protocols with respect to use of random nonces is beyond the scope of this brief contribution and subject to future work, there are at least two examples. In OCRA (OATH Challenge-Response Algorithm), a randomly generated challenge value is exchanged [7]. In HTTP/1.1 digest access authentication, the server sends a 401 response to the client, which includes a nonce [9]. While we could not find published exploits of the above weakness in these protocols, and did not check relevance nor bandwidth, future research is necessary to check if these protocols are indeed affected, and if so, if the same cure as for TLS can also be applied to these protocols.

Moreover, consideration of an update of the TLS protocol specification by standardization bodies would be wishful in order to prevent this type of exploit at least for this widely used protocol in the future.

Acknowledgments

This work is supported by project SIMARGL (Secure intelligent methods for advanced recognition of malware, stegomalware & information hiding methods, https://simargl.eu), which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833042.

REFERENCES

[1] David Chaum. 2004. Secret-Ballot Receipts: True Voter-Verifiable Elections. *IEEE Secur. Priv.* 2, 1 (2004), 38–47. https://doi.org/10.1109/MSECP.2004.1264852

[2] Zbigniew Golebiewski, Miroslaw Kutylowski, and Filip Zagórski. 2006. Stealing Secrets with SSL/TLS and SSH - Kleptographic Attacks. In *Cryptology and Network Security, 5th International Conference, CANS 2006, Suzhou, China, December 8-10, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 4301)*. Springer, Berlin, 191–202. https://doi.org/10.1007/11935070_13

[3] Adam Janovsky, Jan Krhovjak, and Vashek Matyas. 2018. Bringing Kleptography to Real-World TLS. In *Information Security Theory and Practice - 12th IFIP WG 11.2 International Conference, WISTP 2018, Brussels, Belgium, December 10-11, 2018, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 11469)*. Springer, Berlin, 15–27. https://doi.org/10.1007/978-3-030-20074-9_3

[4] Butler Lampson. 1973. A Note on the Confinement Problem. *Commun. ACM* 16, 10 (1973), 613–615. https://doi.org/10.1145/362375.362389

[5] Wojciech Mazurczyk, Steffen Wendzel, Sebastian Zander, Amir Houmansadr, and Krzysztof Szczypiorski. 2016. *Information Hiding in Communication Networks: Fundamentals, Mechanisms, Applications, and Countermeasures.* Wiley-IEEE Press, Hoboken, NJ.

[6] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. 1996. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, FL.

[7] D. M'Raihi, Johan Rydell, Siddharth Bajaj, Salah Machani, and David Naccache. 2011. OCRA: OATH Challenge-Response Algorithm. RFC 6287. https://doi.org/10.17487/RFC6287

[8] Eric Rescorla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. https://doi.org/10.17487/RFC8446

[9] Rifaat Shekh-Yusef, David Ahrens, and Sophie Bremer. 2015. HTTP Digest Access Authentication. RFC 7616. https://doi.org/10.17487/RFC7616

[10] Eric Wustrow, Colleen Swanson, and J. Alex Halderman. 2014. TapDance: End-to-Middle Anticensorship without Flow Blocking. In *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014.* USENIX Association, Dover, DE, 159–174. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/wustrow

[11] Adam L. Young and Moti Yung. 1997. Kleptography: Using Cryptography Against Cryptography. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding (Lecture Notes in Computer Science, Vol. 1233)*. Springer, Berlin, 62–74. https://doi.org/10.1007/3-540-69053-0_6