THE SALSA20 STREAM CIPHER FOR TRANSPORT LAYER SECURITY

S. Josefsson & J. Strombergson & N. Mavrogiannopoulos
SJD Secworks KU Leuven

Overview

- Issues with TLS
- Proposal
- Performance comparison
- Open-questions in the proposal

Issues with TLS

- If the CBC-ciphersuites are implemented by the book/RFC are vulnerable to attacks [0,1]

Issues with TLS

- If the CBC-ciphersuites are implemented by the book/RFC are vulnerable to attacks [0,1]
- There are known attacks in RC4 that cannot be mitigated [2]

Issues with TLS

- If the CBC-ciphersuites are implemented by the book/RFC are vulnerable to attacks [0,1]
- There are known attacks in RC4 that cannot be mitigated [2]

[0]. AlFardan, N., and Paterson, K. "Plaintext-recovery attacks against datagram TLS." In Network and Distributed System Security Symposium (2012).
[1]. AlFardan, Nadhem J., and Kenneth G. Paterson. "Lucky thirteen: Breaking the TLS and DTLS record protocols." IEEE Symposium on Security and Privacy. 2013.
[2]. Isobe, T., Ohigashi, T., Watanabe, Y., and Morii, M., "Full Plaintext Recovery Attack on Broadcast RC4." International Workshop on Fast Software Encryption , 2013.

Issues with TLS

- RC4 cannot be used in Datagram TLS (DTLS)

Issues with TLS

- RC4 cannot be used in Datagram TLS (DTLS)
  - No stream ciphers in DTLS

Issues with TLS: result

- That leaves us with few options
  - AES-GCM
    - Very fast on certain CPUs
    - Decent performance otherwise

Issues with TLS: result

- That leaves us with few options
  - AES-GCM
    - Very fast on certain CPUs
    - Decent performance otherwise
  - AES-CCM
    - Decent performance

Issues with TLS: result

- That leaves us with few options
  - AES-GCM
    - Very fast on certain CPUs
    - Decent performance otherwise
  - AES-CCM
    - Decent performance
  - Both are only applicable to TLS 1.2+ or DTLS 1.2+

Issues with TLS: result

- When decent performance isn't enough, a fast and secure stream cipher is needed

Proposal

- We propose to use the eStream [0] results to define a fast stream cipher for TLS/DTLS

[0]. The eSTREAM project was a multi-year effort, running from 2004 to 2008, to promote the design of efficient and compact stream ciphers suitable for widespread adoption. As a result of the project, a portfolio of stream ciphers was announced in April 2008 and revised in 2012.

Proposal

- We propose to use the eStream [0] results to define a fast stream cipher for TLS/DTLS
  - ESTREAM-SALSA20-HMAC-SHA1
  - SALSA20-HMAC-SHA1

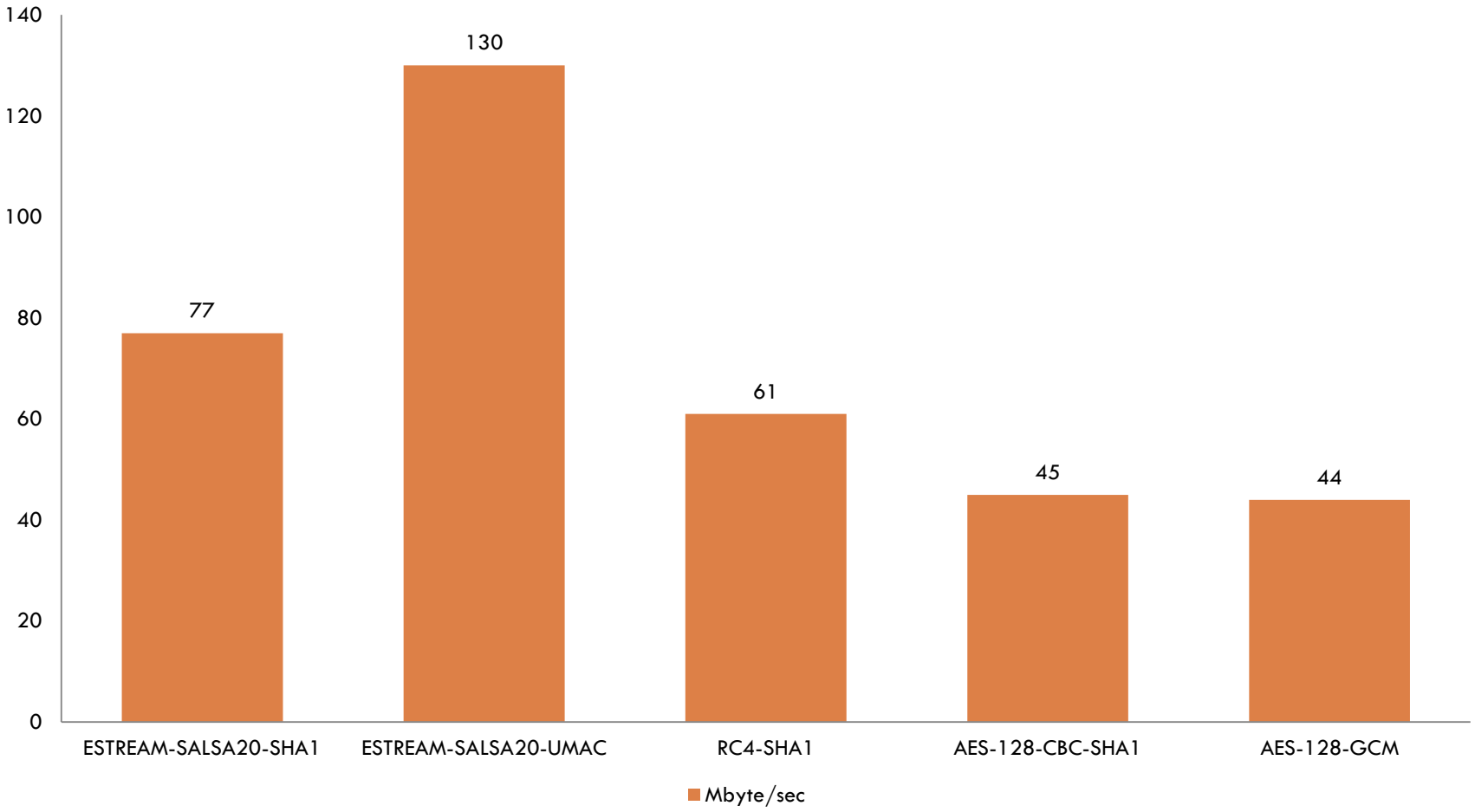
Proposal

- We propose to use the eStream [0] results to define a fast stream cipher for TLS/DTLS
  - ESTREAM-SALSA20-HMAC-SHA1
  - SALSA20-HMAC-SHA1
- and also utilize a fast MAC algorithm
  - ESTREAM-SALSA20-UMAC
  - SALSA20-UMAC

Proposal

- We propose to use the eStream [0] results to define a fast stream cipher for TLS/DTLS
  - ESTREAM-SALSA20-HMAC-SHA1
  - SALSA20-HMAC-SHA1
- and also utilize a fast MAC algorithm
  - ESTREAM-SALSA20-UMAC
  - SALSA20-UMAC
- UMAC as in RFC4418 (UMAC-AES)

Performance comparison



Packet Overhead

- Packet overhead per ciphersuite (in DTLS):

Ciphersuite	Overhead	% of 1500	Expanded
AES-128-CBC-HMAC-SHA1	50-65	3.3-4.3	13 + 20 (MAC) + 16 (IV) + 16 (PAD)
AES-128-GCM	37	2.4	13 + 16 (MAC) + 8 (IV)
SALSA20-256-HMAC-SHA1	33	2.2	13 + 20 (MAC)
SALSA20-256-UMAC96	25	1.6	13 + 12 (MAC)

Packet Overhead

- Packet overhead per ciphersuite (in DTLS):

Ciphersuite	Overhead	% of 1500	Expanded
AES-128-CBC-HMAC-SHA1	50-65	3.3-4.3	13 + 20 (MAC) + 16 (IV) + 16 (PAD)
AES-128-GCM	37	2.4	13 + 16 (MAC) + 8 (IV)
SALSA20-256-HMAC-SHA1	33	2.2	13 + 20 (MAC)
SALSA20-256-UMAC96	25	1.6	13 + 12 (MAC)

the packet counter is the nonce

Open-questions in proposal

- UMAC can be used
  - As in RFC4418 (UMAC-AES)
  - Or with the combined cipher (i.e., Salsa20)

Open-questions in proposal

- UMAC can be used:
  - As in RFC4418 (UMAC-AES)
  - Or with the combined cipher (i.e., Salsa20)
- Poly1305 is another option for a MAC
  - With comparable speed
  - Proposed in 2005 (UMAC in 1999)
  - No RFC

Conclusion

- We can have a replacement of RC4 that is:
  - More secure (one of the winners in eStream competition)
  - Faster
    - 2x-3x the speed of AES ciphersuites
    - 2x the speed of RC4 when combined with UMAC
  - Can be used efficiently with DTLS

Questions and Discussion

Salsa20 cryptanalysis

- Aumasson, Jean-Philippe, et al. "**New features of Latin dances: analysis of Salsa, ChaCha, and Rumba.**" Fast Software Encryption. Springer Berlin Heidelberg, 2008.
- Fischer, Simon, et al. "**Non-randomness in eSTREAM Candidates Salsa20 and TSC-4."** Progress in Cryptology-INDOCRYPT 2006. Springer Berlin Heidelberg, 2006. 2-16.
- Priemuth-Schmid, Deike, and Alex Biryukov. "**Slid pairs in Salsa20 and Trivium.**" Progress in Cryptology-INDOCRYPT 2008. Springer Berlin Heidelberg, 2008. 1-14.
- Hernandez-Castro, Julio Cesar, Juan ME Tapiador, and Jean-Jacques Quisquater. "**On the Salsa20 core function.**" Fast Software Encryption. Springer Berlin Heidelberg, 2008.
- Crowley, Paul. "**Truncated differential cryptanalysis of five rounds of Salsa20.**" IACR Cryptology ePrint Archive 2005 (2005): 375.
- Shao, Zeng-yu, and Lin Ding. "**Related-Cipher Attack on Salsa20.**" Computational and Information Sciences (ICCIS), 2012 Fourth International Conference on. IEEE, 2012.
- Shi, Zhenqing, et al. "**Improved key recovery attacks on reduced-round salsa20 and chacha.**" Information Security and Cryptology–ICISC 2012. Springer Berlin Heidelberg, 2013. 337-351.
- Tsunoo, Yukiyasu, et al. "**Differential cryptanalysis of Salsa20/8.**" Workshop Record of SASC. 2007.
- Mouha, Nicky, and Bart Preneel. "**A Proof that the ARX Cipher Salsa20 is Secure against Differential Cryptanalysis.**"
- Pelissier, Sylvain. "**Cryptanalysis of Reduced Word Variants of Salsa.**" Western European Workshop on Research in Cryptology, WEWoRC. Vol. 44. 2009."
- Estream portfolio page for Salsa20: http://www.ecrypt.eu.org/stream/e2-salsa20.html

Performance comparison (full)