리눅스 커널 오픈소스와 Kakao i Cloud

유태희
카카오엔터프라이즈

01 Linux Kernel

- Kakao i Cloud가 사용중인 오픈소스들
  - Linux Kernel
  - Openstack
  - Kubernetes
  - etc

01 Linux Kernel

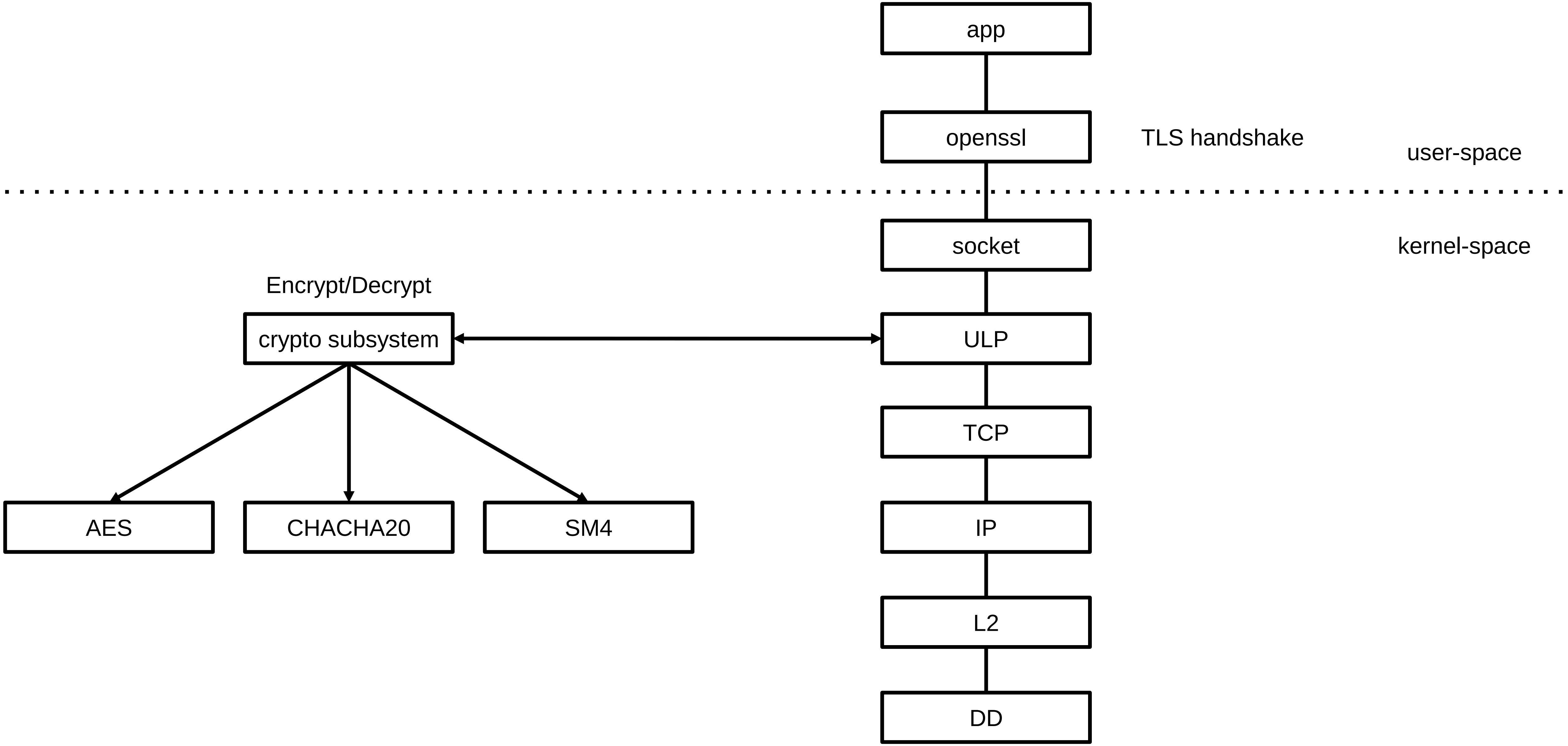
- FPGA
  - SmartNIC
- CPU
  - SIMD instruction
  - AVX, AVX2, AVX512

01 Linux Kernel

- SIMD in Linux kernel
- Crypto algorithms
  - AES, CAMELLIA, SHA, etc
- Netfilter
- raid6

02 ARIA-kTLS

- kTLS
  - In kernel TLS implementation
  - No handshake
  - Encryption / Decryption only


app
openssl
TLS handshake
user-space
socket
kernel-space
Encrypt/Decrypt
crypto subsystem
ULP
TCP
AES
CHACHA20
SM4
IP
L2
DD

02 ARIA-kTLS

- kTLS
  - AES, CHACHA20, SM4 알고리즘 지원
  - user-space대비 9%성능 향상
  - hardware offload interface제공

03 ARIA-kTLS

- ARIA
  - RFC 5794, 국제표준
  - 블록 암호화(128bit)
  - 128, 192, 256bit key
  - OpenSSL에 구현되어있음.

Independent Submission
Request for Comments: 5794
Category: Informational
ISSN: 2070-1721

J. Lee
J. Lee
J. Kim
D. Kwon
C. Kim
NSRI

March 2010

A Description of the ARIA Encryption Algorithm

Abstract

This document describes the ARIA encryption algorithm. ARIA is a 128-bit block cipher with 128-, 192-, and 256-bit keys. The algorithm consists of a key scheduling part and data randomizing part.

Internet Engineering Task Force (IETF)
Request for Comments: 6209
Category: Informational
ISSN: 2070-1721

W. Kim
J. Lee
J. Park
D. Kwon
NSRI

April 2011

Addition of the ARIA Cipher Suites to Transport Layer Security (TLS)

Abstract

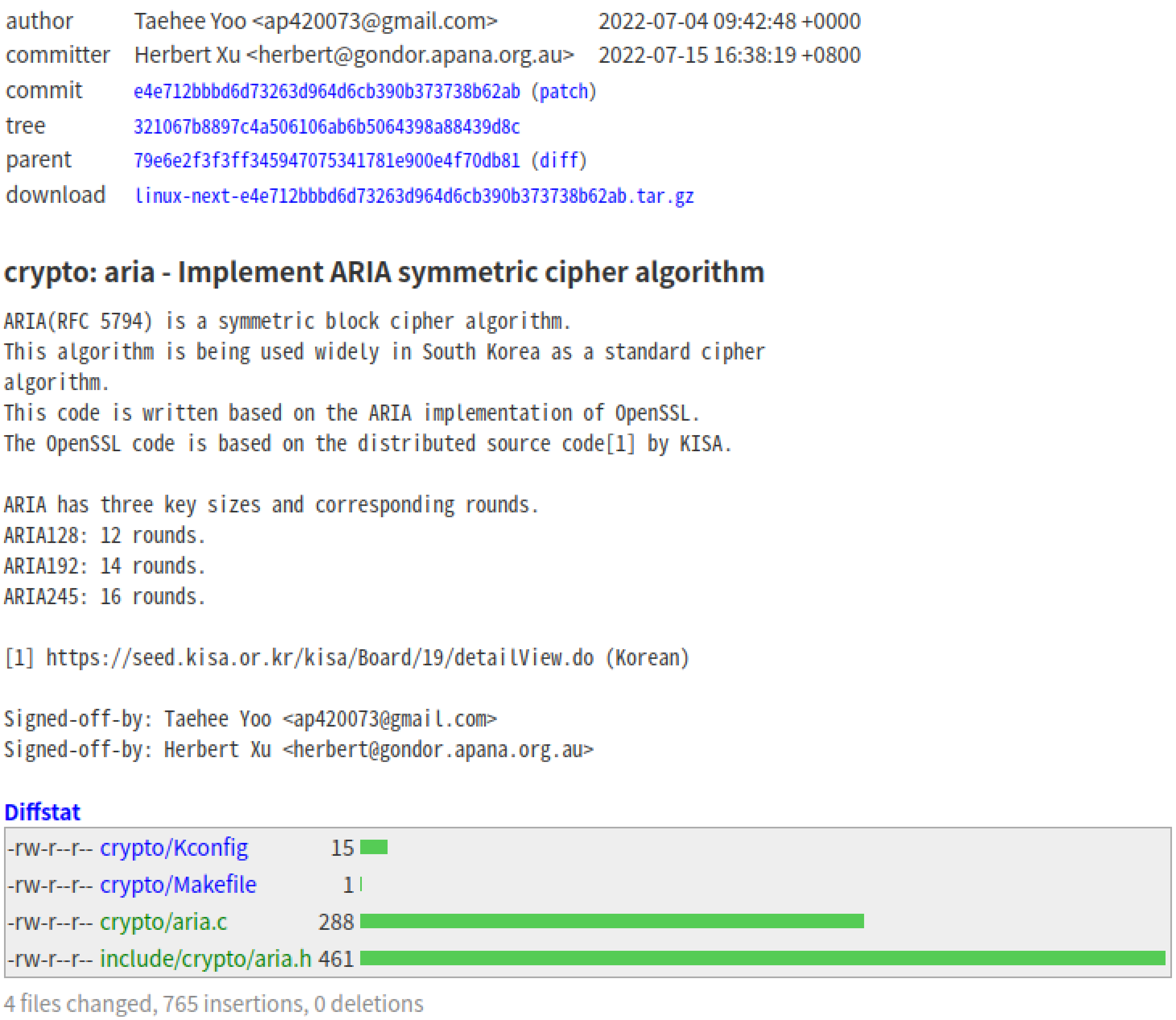
This document specifies a set of cipher suites for the Transport Layer Security (TLS) protocol to support the ARIA encryption algorithm as a block cipher.

02 ARIA-kTLS

- 목적
  - kTLS에서 ARIA알고리즘을 사용할 수 있도록 함.
  - 성능적인 이점 있음.
  - 국내 ARIA 사용자들에게 이득이 있을것으로 예상.
- 단계
  - ARIA알고리즘을 kernel에 merge
  - kTLS에 ARIA설정 부분 추가
  - ARIA-AVX구현

02 ARIA-kTLS

- ARIA구현
  - 8bit, 32bit구현체 존재
  - KISA에서 ARIA구현체 배포중
  - https://seed.kisa.or.kr/kisa/Board/19/detailView.do
  - OpenSSL에 구현되어 있음.



author	Taehee Yoo <ap420073@gmail.com>	2022-07-04 09:42:48 +0000
committer	Herbert Xu <herbert@gondor.apana.org.au>	2022-07-15 16:38:19 +0800
commit	e4e712bbbd6d73263d964d6cb390b373738b62ab (patch)	
tree	321067b8897c4a506106ab6b5064398a88439d8c	
parent	79e6e2f3f3ff345947075341781e900e4f70db81 (diff)	
download	linux-next-e4e712bbbd6d73263d964d6cb390b373738b62ab.tar.gz	

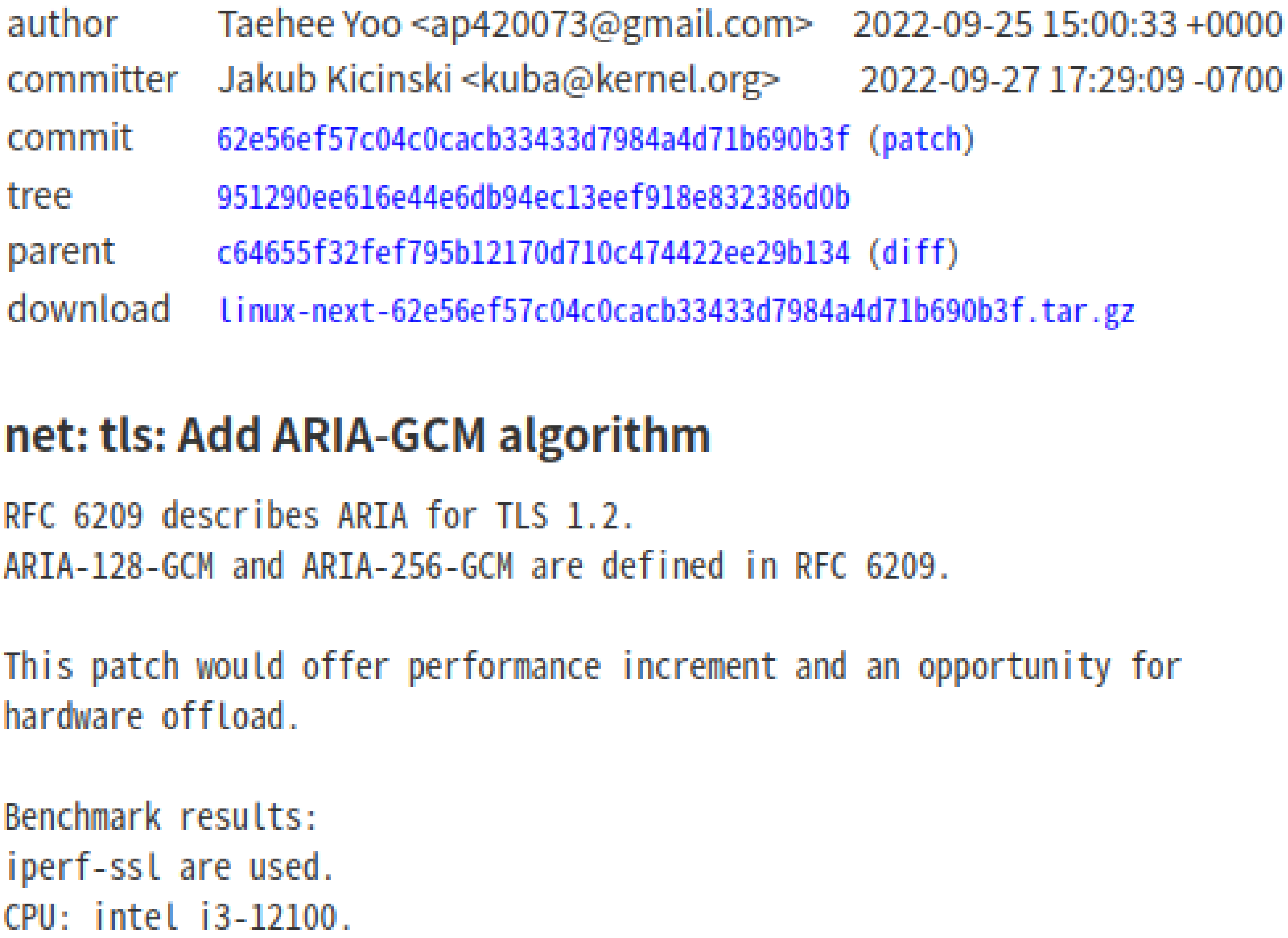
crypto: aria - Implement ARIA symmetric cipher algorithm

```
ARIA(RFC 5794) is a symmetric block cipher algorithm.
This algorithm is being used widely in South Korea as a standard cipher
algorithm.
This code is written based on the ARIA implementation of OpenSSL.
The OpenSSL code is based on the distributed source code[1] by KISA.

ARIA has three key sizes and corresponding rounds.
ARIA128: 12 rounds.
ARIA192: 14 rounds.
ARIA245: 16 rounds.

[1] https://seed.kisa.or.kr/kisa/Board/19/detailView.do (Korean)

Signed-off-by: Taehee Yoo <ap420073@gmail.com>
Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>
```

Diffstat

-rw-r--r--	crypto/Kconfig	15
-rw-r--r--	crypto/Makefile	1
-rw-r--r--	crypto/aria.c	288
-rw-r--r--	include/crypto/aria.h	461

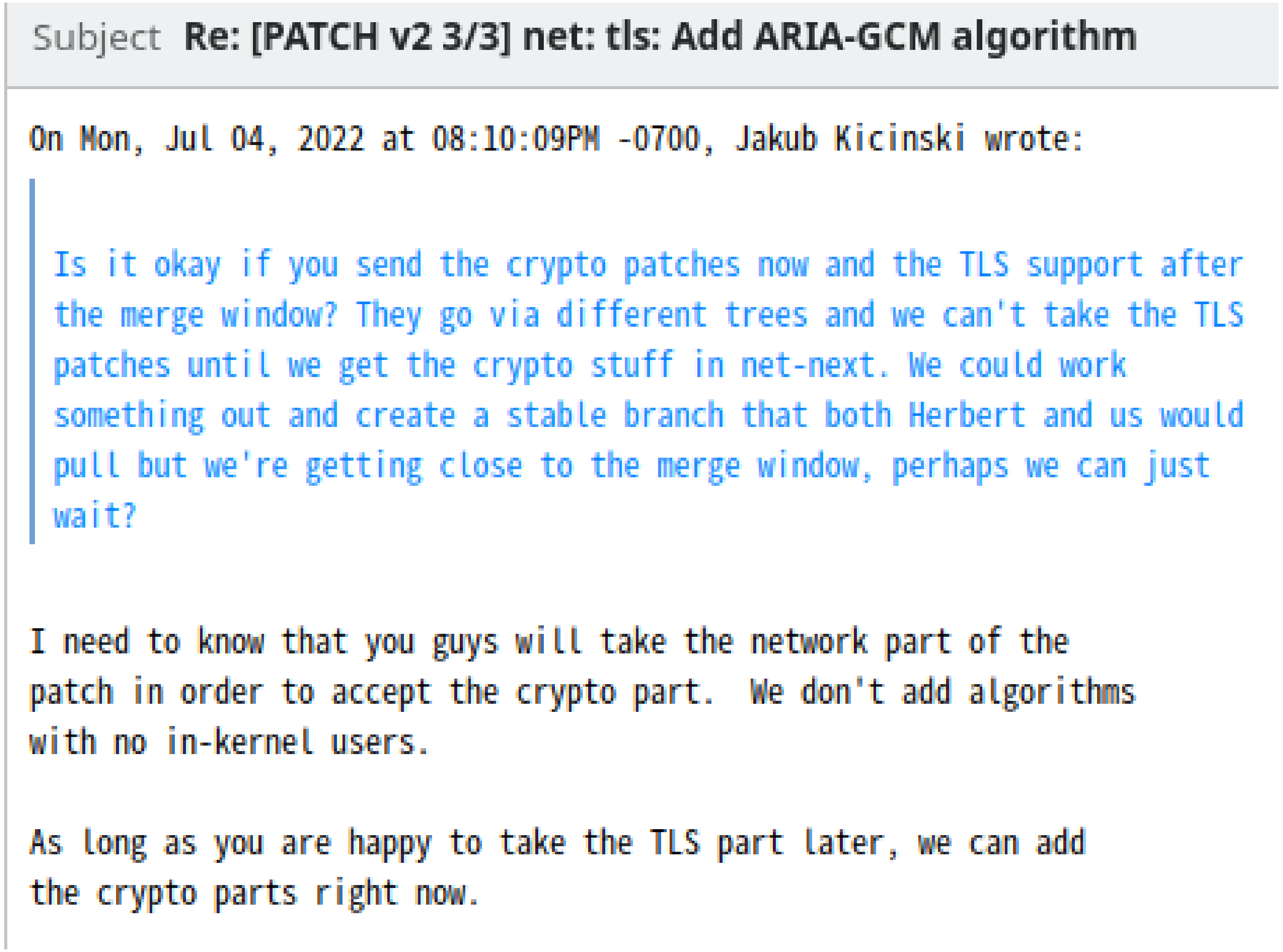
4 files changed, 765 insertions, 0 deletions



author	Taehee Yoo <ap420073@gmail.com>	2022-09-25 15:00:33 +0000
committer	Jakub Kicinski <kuba@kernel.org>	2022-09-27 17:29:09 -0700
commit	62e56ef57c04c0cacb33433d7984a4d71b690b3f (patch)	
tree	951290ee616e44e6db94ec13eef918e832386d0b	
parent	c64655f32fef795b12170d710c474422ee29b134 (diff)	
download	linux-next-62e56ef57c04c0cacb33433d7984a4d71b690b3f.tar.gz	

net: tls: Add ARIA-GCM algorithm

```
RFC 6209 describes ARIA for TLS 1.2.
ARIA-128-GCM and ARIA-256-GCM are defined in RFC 6209.

This patch would offer performance increment and an opportunity for
hardware offload.

Benchmark results:
iperf-ssl are used.
CPU: intel i3-12100.
```



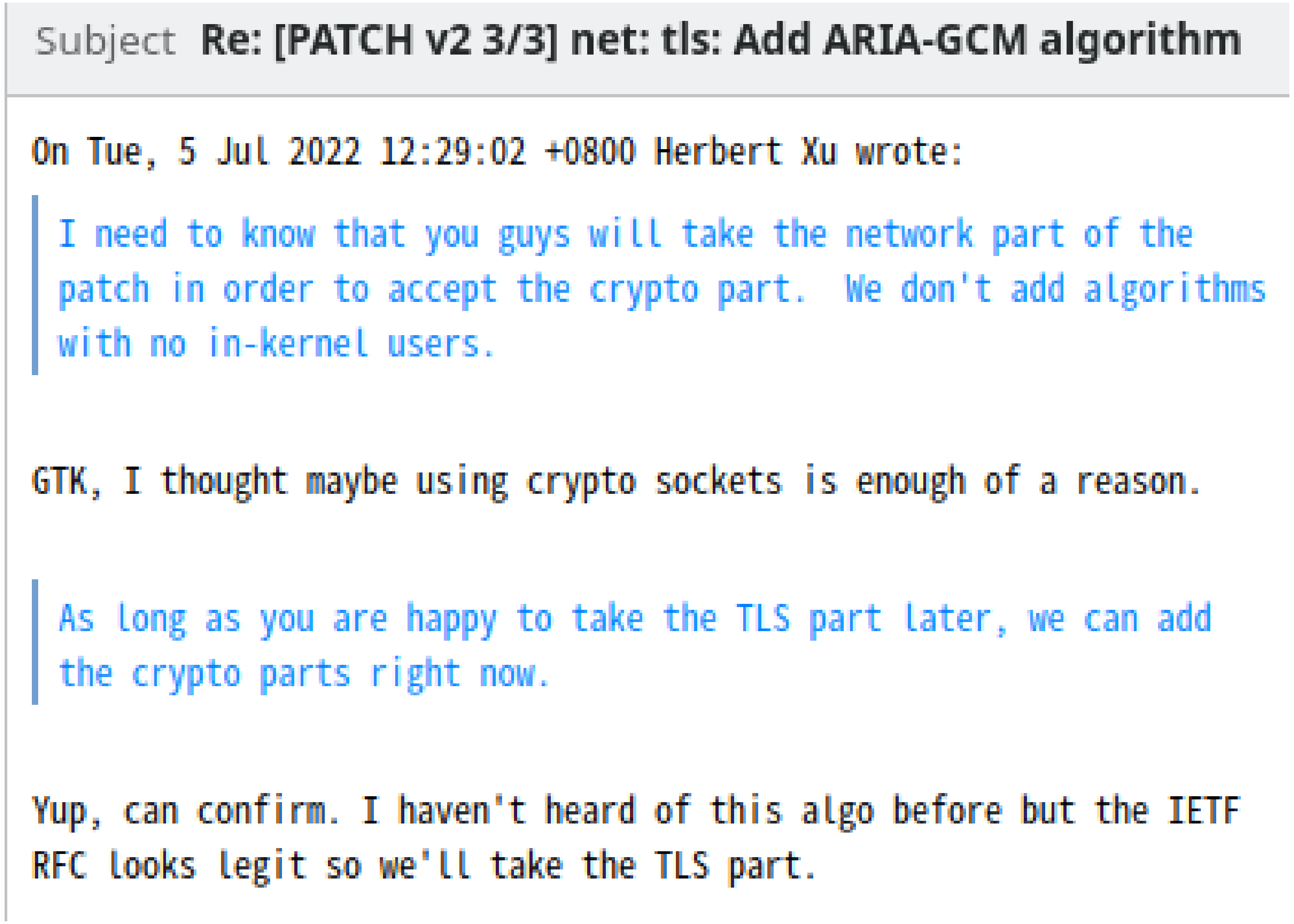
Subject **Re: [PATCH v2 3/3] net: tls: Add ARIA-GCM algorithm**

On Mon, Jul 04, 2022 at 08:10:09PM -0700, Jakub Kicinski wrote:

> Is it okay if you send the crypto patches now and the TLS support after the merge window? They go via different trees and we can't take the TLS patches until we get the crypto stuff in net-next. We could work something out and create a stable branch that both Herbert and us would pull but we're getting close to the merge window, perhaps we can just wait?

I need to know that you guys will take the network part of the patch in order to accept the crypto part. We don't add algorithms with no in-kernel users.

As long as you are happy to take the TLS part later, we can add the crypto parts right now.



Subject **Re: [PATCH v2 3/3] net: tls: Add ARIA-GCM algorithm**

On Tue, 5 Jul 2022 12:29:02 +0800 Herbert Xu wrote:

> I need to know that you guys will take the network part of the patch in order to accept the crypto part. We don't add algorithms with no in-kernel users.

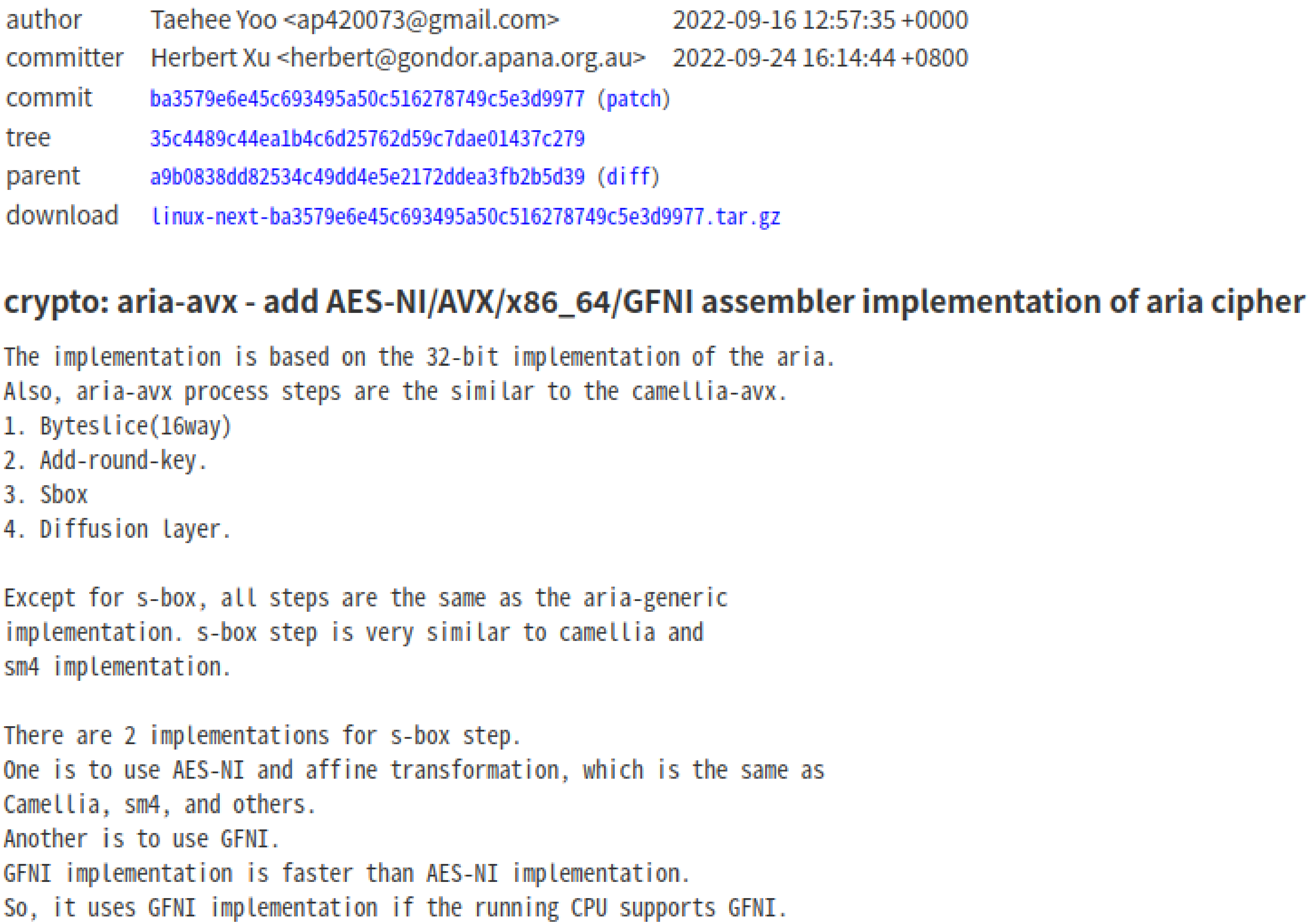
GTK, I thought maybe using crypto sockets is enough of a reason.

> As long as you are happy to take the TLS part later, we can add the crypto parts right now.

Yup, can confirm. I haven't heard of this algo before but the IETF RFC looks legit so we'll take the TLS part.

02 ARIA-kTLS

- AVX + AES-NI/GFNI
  - 128bit
  - 16way parallel
  - generic(c언어 사용)구현 대비 3~4배 성능 향상
- AVX2 + AES-NI/GFNI
  - 256bit
  - 32way parallel
  - generic대비 7배 성능 향상
- AVX512 + GFNI
  - 512bit
  - 64way parallel
  - generic대비 10~11배 성능 향상



author	Taehee Yoo <ap420073@gmail.com>	2022-09-16 12:57:35 +0000
committer	Herbert Xu <herbert@gondor.apana.org.au>	2022-09-24 16:14:44 +0800
commit	ba3579e6e45c693495a50c516278749c5e3d9977 (patch)	
tree	35c4489c44ea1b4c6d25762d59c7dae01437c279	
parent	a9b0838dd82534c49dd4e5e2172ddea3fb2b5d39 (diff)	
download	linux-next-ba3579e6e45c693495a50c516278749c5e3d9977.tar.gz	

crypto: aria-avx - add AES-NI/AVX/x86_64/GFNI assembler implementation of aria cipher

```
The implementation is based on the 32-bit implementation of the aria.
Also, aria-avx process steps are the similar to the camellia-avx.
1. Byteslice(16way)
2. Add-round-key.
3. Sbox
4. Diffusion layer.

Except for s-box, all steps are the same as the aria-generic
implementation. s-box step is very similar to camellia and
sm4 implementation.

There are 2 implementations for s-box step.
One is to use AES-NI and affine transformation, which is the same as
Camellia, sm4, and others.
Another is to use GFNI.
GFNI implementation is faster than AES-NI implementation.
So, it uses GFNI implementation if the running CPU supports GFNI.
```

03 결론

- Linux Kernel에서 고성능 ARIA알고리즘을 제공하는것은 국내 유저에게 큰 이득
- 보수적인 crypto subsystem 리뷰어들을 설득시킬 수 있는것은 확실한 use-case 및 국제표준.
- ARIA-TLS(RFC 6209)표준화가 큰 도움.
- ARIA-IPSEC표준화가 되지 못한 부분은 큰 아쉬움.

감사합니다

리눅스 커널 오픈소스와 Kakao i Cloud