IETFにおける暗号アルゴリズムを用いた通信プロトコルの現状と今後の展望

菅野 哲
株式会社レピダム

lepidum
https://lepidum.co.jp/

CRYPTRECシンポジウム2017

本講演の聞きどころ？！

＜講演タイトル＞

IETFにおける暗号アルゴリズムを用いた通信プロトコルの現状と今後の展望

＜本講演で知って･考えてもらいたいこと＞

- IETFという標準化団体の存在と役割
- IETFでの暗号技術の採用状況とその役割
- CRYPTRECとして何ができるのか

本日のアジェンダ

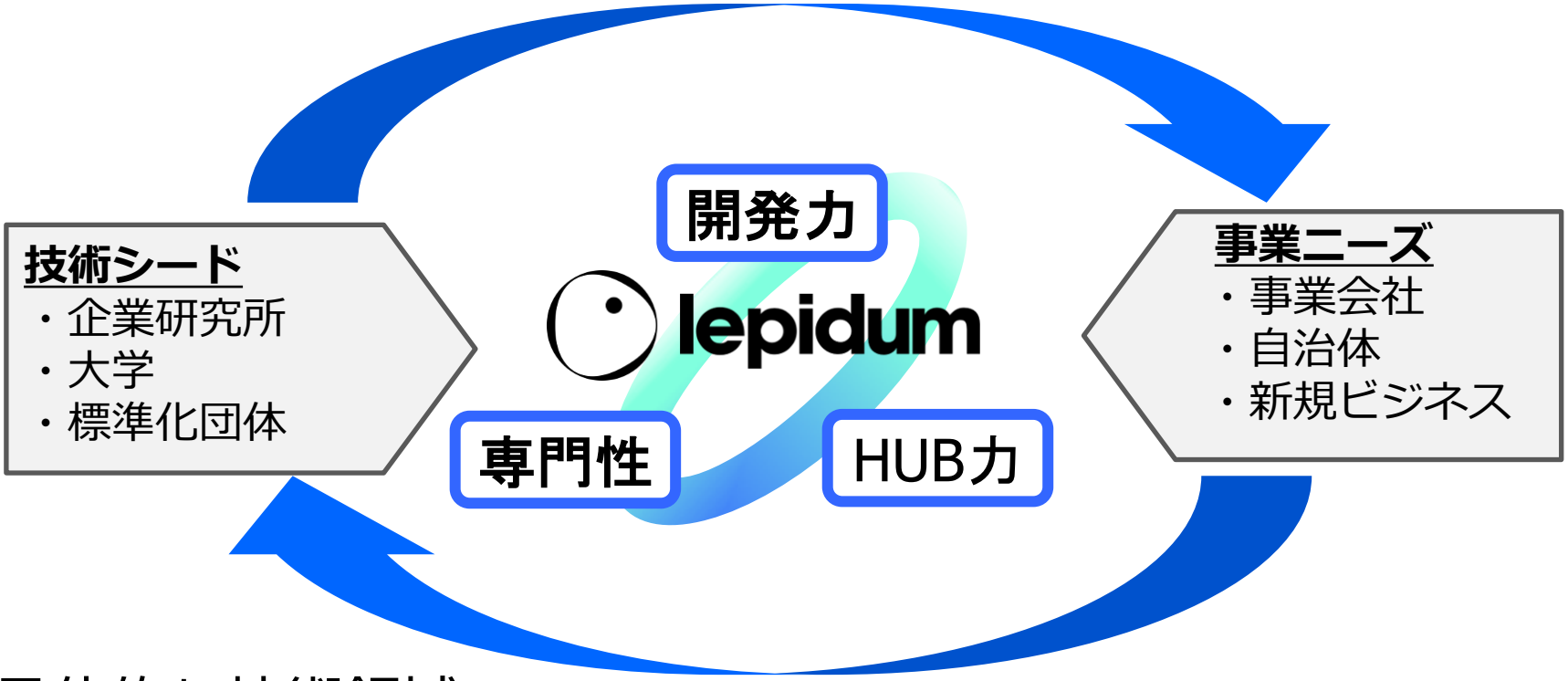
- 準備
  - IETFとは
- 本題
  - IETFと暗号アルゴリズムの関係
  - IETFでの暗号が使われている通信プロトコル
  - 通信プロトコルの現状
    - 具体例：SSL/TLS
    - 暗号技術の側面からCRYPTRECとIETFを比較
  - 今後の展望
- 我々はどうすべきか
  - 暗号利用の力学が変わってきているのでは？
  - インターネットでの暗号技術の利用に追従するには・・
  - CRYPTRECに期待すること
- まとめ

この人、誰？

- 名前
  - 菅野 哲（かんの さとる）
- 所属
  - 株式会社 レピダム 取締役
  - ISOC-JP プログラム委員
- どんなことやっていた／やっているの？
  - 学生時代～
    - 暗号プロトコルの研究、暗号製品を売り歩く（？）
  - 社会人～
    - 暗号ライブラリや情報セキュリティ関連システム開発
    - Camelliaに関する標準化活動
      - RFCを何本か発行
    - 人事部で人材開発
  - （ここ最近）
    - 会社に関することは何でも！？

株式会社レピダムって・・・？

エッジの効いた技術でお客様の事業を加速させる燃料

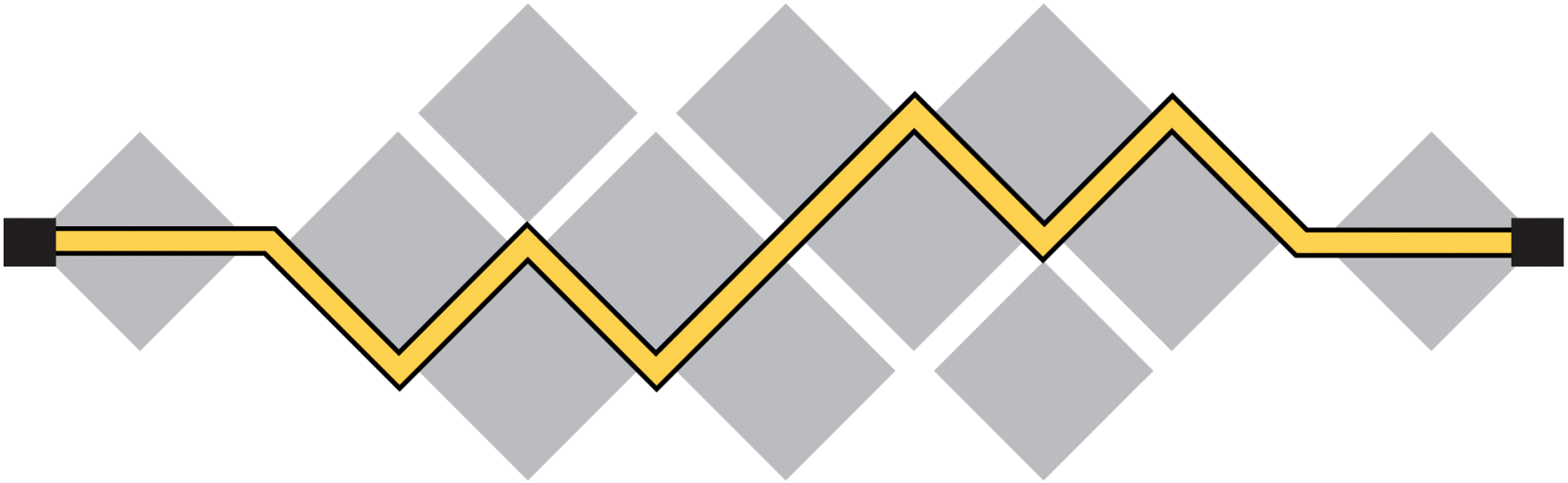


具体的な技術領域：

「標準化支援、アイデンティティ、プライバシ、認証・認可、情報セキュリティ」に関する開発、調査・コンサル

本日のアジェンダ

- 準備
  - IETFとは
- 本題
  - IETFと暗号アルゴリズムの関係
  - IETFでの暗号が使われている通信プロトコル
  - 通信プロトコルの現状
    - 具体例：SSL/TLS
    - 暗号技術の側面からCRYPTRECとIETFを比較
  - 今後の展望
- 我々はどうすべきか
  - 暗号利用の力学が変わってきているのでは？
  - インターネットでの暗号技術の利用に追従するには・・
  - CRYPTRECに期待すること
- まとめ

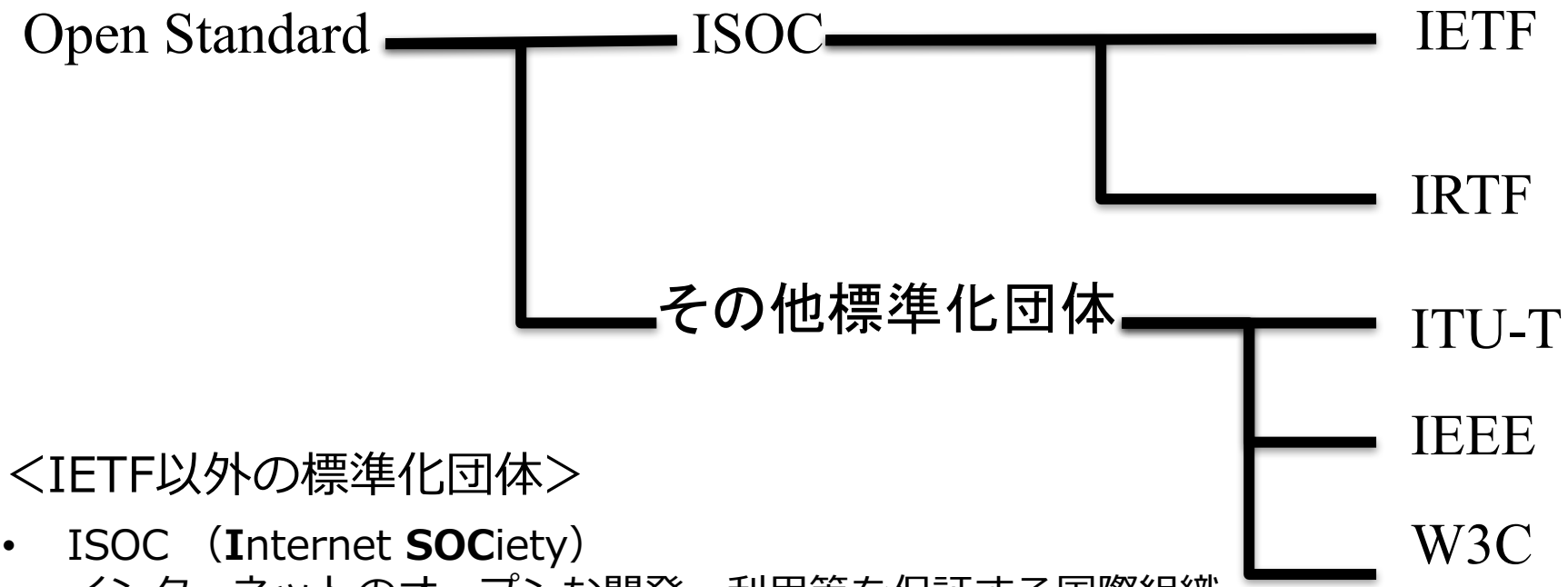


IETF®

IETFとは（1/5）

- **I**nternet **E**ngineering **T**ask **F**orce
  - **インターネットに関する技術**の国際標準を策定する組織
- 理念
  - “We reject kings, presidents and voting. We believe in ***rough consensus*** and ***running code***.” David Clark (1992)
- 生産物
  - RFC （Request for Comments）を発行
    - インターネットを技術的な側面を支えられている
- 活動
  - 年3回開催（3月、7月、11月）で１週間
  - 参加者数：1000～1500人
  - 参加費：700USD
  - 参加資格：誰でもOK

IETFとは（2/5）



＜IETF以外の標準化団体＞

- ISOC （**I**nternet **SOC**iety）
  インターネットのオープンな開発・利用等を保証する国際組織
- IRTF （**I**nternet **R**esearch **T**ask **F**orce） ：
  インターネットの未来において重要と思われる研究を推進する組織
- ITU-T：
  国際電気通信連合において通信分野の標準化策定を担当する電気通信標準化部門
- IEEE：
  アメリカに本部を持つ電気電子技術学会
- W3C：
  World Wide Webで使用される各種技術の標準化の推進を目的に設立された団体

補足情報1：標準化団体の比較

標準化スタイルはISOはトップダウン、IETFはボトムアップ

トップダウン

ISO	IETF
Specification Oriented （まず仕様を決める）	Implementation Oriented （まずは実装をする）
Hard Specification （仕様は変わらない）	Soft Specification （仕様は変わっていく）
Quality of Service	Connectivity
Voting	Running Code & Rough Consensus
Membership	Volunteer

ボトムアップ

補足情報2：標準化の種類

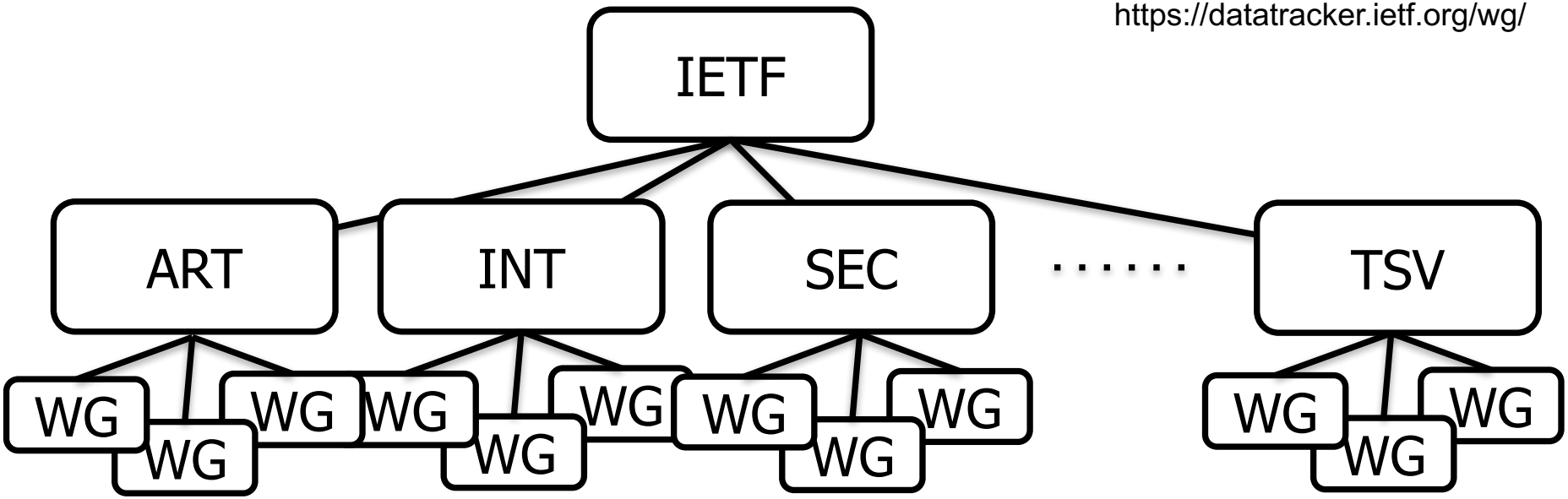
標準化ドキュメントには大きく分けて２種類が存在

- デジュール標準
  - De jure = 法的に正式
  - 国際的な取り決めが必要／正式な会議での議論および決定
  - 例：ISO、ITU、JIS etc.

- デファクト標準
  - De facto = 事実上の標準
  - 製品が広く受け入れられ、事実上の標準
  - 市場ドリブン
  - 例：IETF、W3C etc.

IETFとは（3/5）

https://datatracker.ietf.org/wg/



7 Area
122WGs
(2017.11現在)

- GEN（General）: 1
- ART（Applications and Real-Time）:35
- INT（Internet）: 18
- OPS（Operations and Management）: 16
- RTG（Routing）: 25
- SEC（Security）:15
- TSV（Transport and Services）: 12

補足情報3：エリアと代表的なWG

IETFだけにインターネットに関連した技術の技術的な検討を中心として実施

	エリア	代表的なWG
IETF	GEN	mtgvenue
	ART	httpbis, uta, dcrup etc.
	INT	6man, 6lo, lwig etc.
	OPS	dnsop, v6ops etc.
	RTG	ospf, sidr etc.
	SEC	**tls, ipsecme, curdle etc.**
	TSV	quic, tcpinc etc.
IRTF		**cfrg**, t2trg etc.

IETFとは（4/5）

自由な雰囲気＆やり過ぎな面も？！:)

IETFとは（5/5）

個人が所属や国を超えた活発な議論

IETFで発行されるRFCってなんだろう？

- 気になるポイント
  - どのように発行されるの？（プロセス）
  - どんな種類のドキュメントがあるの？

＜RFCの具体例＞

Internet Engineering Task Force (IETF) S. Kanno
Request for Comments: 6367 NTT Software Corporation
Category: Informational M. Kanda
ISSN: 2070-1721 NTT
September 2011

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

Abstract

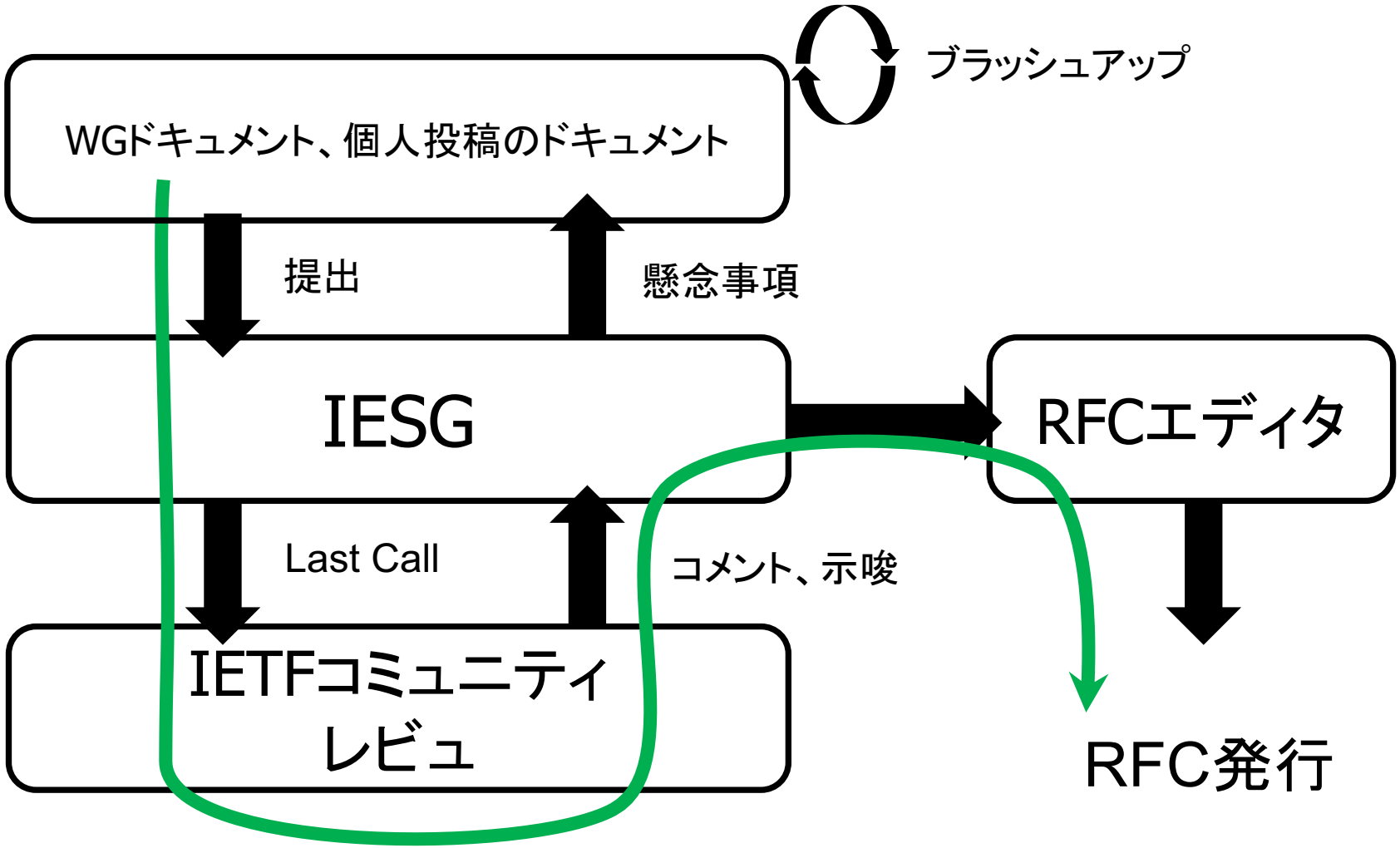
This document specifies forty-two cipher suites for the Transport Security Layer (TLS) protocol to support the Camellia encryption algorithm as a block cipher.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

RFCの標準化プロセス

- RFC発行までの流れ（概略）



RFCの種類

RFCは目的・用途によって以下のようなにカテゴリに分類

- Standard Track
  - Best Current Practice(BCP), Proposed Standard(PS), Internet Standard(STD)の３種類
    - BCP ： 現時点における最良の方法
    - PS ： 良いアィディアであり、問題が報告されていない方法
    - STD ： 上記に加えて、安定＆インターネットに有益な方法
- Informational
  - 業界にとって有益だと考えられる情報であり、情報提供が目的
  - ベンダの独自仕様や特定分野で既にデファクトとして利用されているプロトコルなど
- Experimental
  - 研究や実験目的の扱いとされる方法
- Historical
  - 過去に標準化されたが既に使用されなくなったRFCやその技術がインターネットにとって有害であると判断されたRFC

IETFで標準化された通信プロトコルは？

誰もが聞いたことのある通信プロトコルが多数存在しており、我々の生活を支えている

- アプリケーション
  - DNS、FTP、HTTP、POP3、TELNET、TLS、SSH etc.
- トランスポート
  - TCP、UDP、SCTP etc.
- インターネット
  - ICMP、IPv4、IPv6、ARP etc.
- リンク
  - PPP etc.

ざっくりとIETFをまとめる

- 組織的側面
  - “ラフコンセンサス”と”動作する実装”を武器にインターネットを推進
  - IETFだけでなくIRTFという将来的なことを想定した組織体も存在
- 仕様的側面
  - 変更することが前提の仕様であるRFCによりインターネットを良い方へ先導
  - 用途に適したドキュメントカテゴリで発行
- 影響的側面
  - IETFで標準化された通信プロトコルが身の回りで広く利用

想像しているより現代社会への影響度は大

本日のアジェンダ

- 準備
  - IETFとは
- 本題
  - IETFと暗号アルゴリズムの関係
  - IETFでの暗号が使われている通信プロトコル
  - 通信プロトコルの現状
    - 具体例：SSL/TLS
    - 暗号技術の側面からCRYPTRECとIETFを比較
  - 今後の展望
- 我々はどうすべきか
  - 暗号利用の力学が変わってきているのでは？
  - インターネットでの暗号技術の利用に追従するには・・
  - CRYPTRECに期待すること
- まとめ

IETFと暗号アルゴリズムの関係

インターネットの拡大・成長により暗号技術が重要に！

- 過去（黎明期）
  - セキュリティは不要
    - 特定の個人との接続だったため
- 現在
  - セキュリティは必要というか必須
    - 不特定多数が利用するようになったため

認証やデータの完全性などを実現するために
要素技術として暗号技術の導入へ

IETFでの暗号が使われている通信プロトコル

暗号技術を利用した通信プロトコルは想像以上に多い

- IPsec
- Kerberos
- APOP
- RADIUS
- SSL/TLS
- DTLS
- IKE
- CRAM-MD5
- MIKEY
- CoAP
- など

ある時まで・・・

IETFの黎明期から標準化されているため
レガシー化しているものが多く存在

あの事件を覚えていますか・・・

スノーデン事件・・・

追加

- 2013年6月にエドワード・スノーデン氏によって暴露された一連の出来事
- 情報収集プログラム PRISM（プリズム）
  - グーグルやアップル、フェイスブック、ヤフーなど大手ネット企業が持つデータにアクセス？！

- 事例: 暗号技術への影響
  - 乱数生成アルゴリズム Dual_EC_DRBG (Dual Elliptic Curve Deterministic Random Bit Generator)
    - 2006年 SP800-90Aとして規定
    - 2013年9月 利用しないよう推奨する勧告

Before / After IETF88（1/2）

IETF88 Technical Plenary (Nov, 2013)

2013年6月に明らかになったPRISMによる「Pervasive Surveillance」に対してどうすべきかを検討！

Surveillanceはインターネットへの攻撃と認識！
どのように技術的に対抗すべきか検討する流れ

Before / After IETF88（2/2）

IETF88をきっかけにIETFとしての取り組みに変化が発生

- Before IETF88
  - 暗号技術は必要だけど、ちょっと面倒くさい
  - 今のままでも良いんじゃないかな？
- After IETF88
  - どうやったら身を守れるのか？を真剣に悩む
    - End to End 暗号化を導入すべき！！
  - 暗号アルゴリズムも与えられたものではなく、自分たちで検討したものを選択しよう！
  - オープンな仕様以外のモノは信用できない
    - 例：HSMって危なくないか？という議論

暗号化熱の度が過ぎちゃっている感も・・・

IETFにおける通信プロトコルの現状

“After IETF88”から4年以上経過した現在も
暗号化熱は冷めやらない

- End to End 暗号化を採用
  - Opportunistic encryption でも
- 自分たちで認めた暗号アルゴリズムを採用
- 一時的な（Ephemeral）鍵交換アルゴリズムを推奨
- 各レイヤで暗号化通信のためのプロトコルを検討

これによる弊害も若干出始めているのでは？

SSL/TLSでの暗号技術

暗号利用に関する考え方はRFC7525によって方針確定

- TLS 1.2
  - RFC7525 で推奨設定を策定
    - Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)
  - 暗号設定（推奨）
    - 128ビットセキュリティ以上
    - 鍵交換：DHE/ECDHE（StaticではなくEphemeral）
    - 共通鍵：AES-GCM（AEAD）
- TLS 1.3
  - Internet Draft #21
    - The Transport Layer Security (TLS) Protocol Version 1.3
  - 暗号設定
    - 基本方針はRFC7525に準拠
    - RSA-PSS（署名）やCurve25519、Curve448（楕円曲線）などが追加
    - もちろんChaCha20-Poly1305も追加

“After IETF88” 以降で登場した暗号アルゴリズム 追加

- 共通鍵暗号（ストリーム暗号）
  - ChaCha20
    - 2008年 DJBによって発表
    - Salsa20の変形版
- メッセージ認証コード
  - Poly1305
    - 2005年 DJBによって発表
    - ChaCha20と組合せてデータの秘匿性、完全性、認証性を同時に実現
- 署名アルゴリズム
  - EdDSA（Edwards-curve Digital Signature Algorithm）
    - 2011年 DJBによって発表
    - ツイストしたエドワード曲線に基づくシュノア署名の一種
- 楕円曲線
  - Curve25519 & Curve448
    - 2005年 DJBによって発表
    - この曲線を使ったECDHEはX25519やX448と表現

暗号技術の側面からCRYPTRECとIETFを比較

言わずと知れた「CRYPTREC暗号リスト」

と

IETF curdle WGでの活動

を比較

CRYPTRECでの暗号技術に関する推奨リスト

- CRYPTREC暗号リストは、現在の利用環境と比較するとギャップを感じる（個人的に）

電子政府推奨暗号リスト

技術分類		名称
公開鍵暗号	署名	DSA
		ECDSA
		RSA-PSS(注1)
		RSASSA-PKCS1-v1_5(注1)
	守秘	RSA-OAEP(注1)
	鍵共有	DH
		ECDH
共通鍵暗号	64 ビットブロック暗号(注2)	3-key Triple DES(注3)
	128 ビットブロック暗号	AES
		Camellia
	ストリーム暗号	KCipher-2
ハッシュ関数		SHA-256
		SHA-384
		SHA-512
暗号利用モード	秘匿モード	CBC
		CFB
		CTR
		OFB
	認証付き秘匿モード	CCM
		GCM(注4)
メッセージ認証コード		CMAC
		HMAC
エンティティ認証		ISO/IEC 9798-2
		ISO/IEC 9798-3

推奨候補暗号リスト

技術分類		名称
公開鍵暗号	署名	該当なし
	守秘	該当なし
	鍵共有	PSEC-KEM(注5)
共通鍵暗号	64 ビットブロック暗号(注6)	CIPHERUNICORN-E
		Hierocrypt-L1
		MISTY1
	128 ビットブロック暗号	CIPHERUNICORN-A
		CLEFIA
		Hierocrypt-3
		SC2000
	ストリーム暗号	Enocoro-128v2
		MUGI
		MULTI-S01(注7)
ハッシュ関数		SHA-512/256
		SHA3-256
		SHA3-384
		SHA3-512
		SHAKE128(注12)
		SHAKE256(注12)
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード	該当なし
メッセージ認証コード		PC-MAC-AES
エンティティ認証		ISO/IEC 9798-4

運用監視暗号リスト

技術分類		名称
公開鍵暗号	署名	該当なし
	守秘	RSAES-PKCS1-v1_5(注8)(注9)
	鍵共有	該当なし
共通鍵暗号	64 ビットブロック暗号	該当なし
	128 ビットブロック暗号	該当なし
	ストリーム暗号	128-bit RC4(注10)
ハッシュ関数		RIPEMD-160
		SHA-1(注8)
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード	該当なし
メッセージ認証コード		CBC-MAC(注11)
エンティティ認証		該当なし

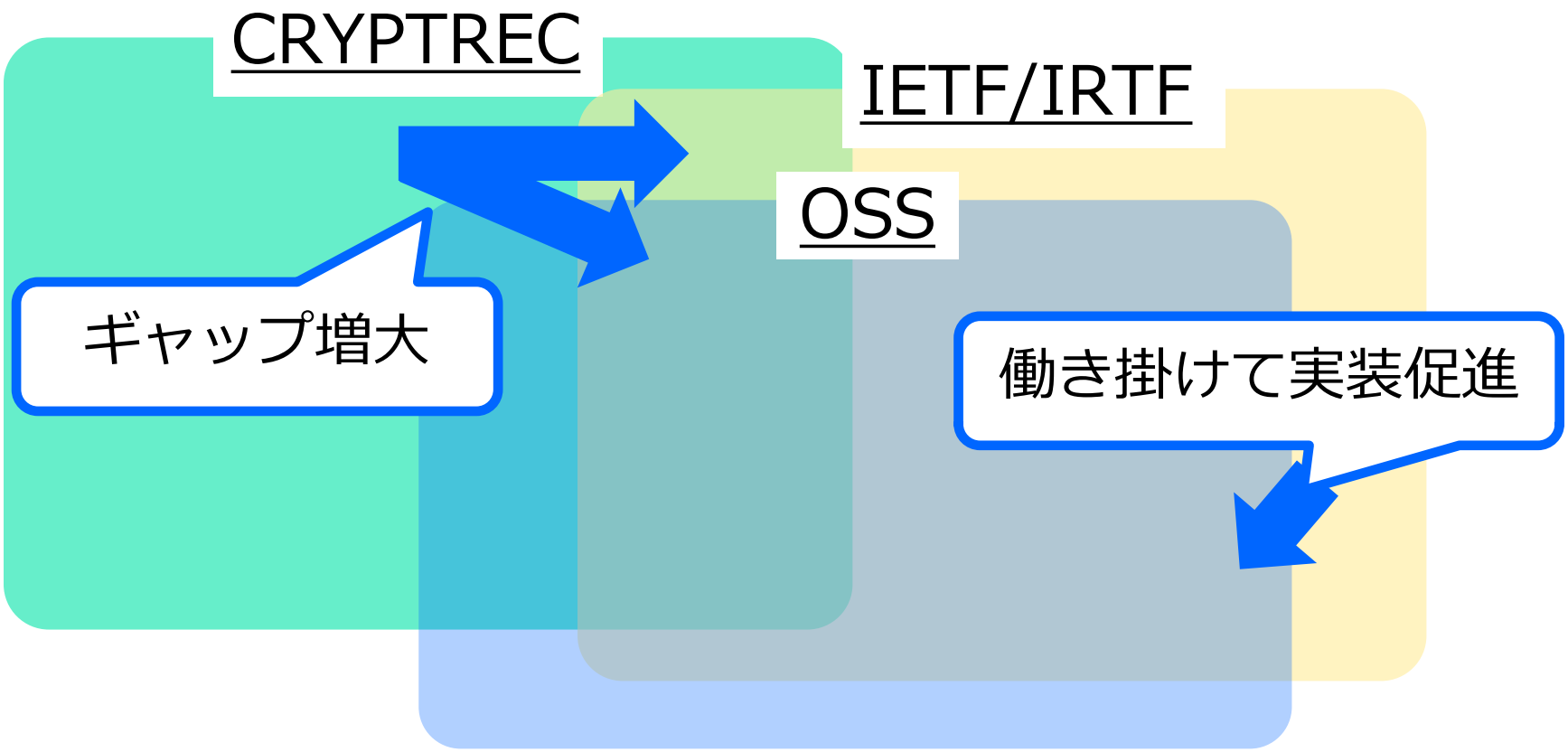
http://www.cryptrec.go.jp/list/cryptrec-ls-0001-2016.pdf

IETFでの暗号技術の推進に関する動き

- IETFでの推奨暗号技術を推進： curdle WG
  - https://datatracker.ietf.org/wg/curdle/about/
- このWGの目的は？
  - 古く危殆化した暗号アルゴリズムの利用停止
  - IRTF cfrgで決めた暗号アルゴリズムへのアップデート
- 具体的な活動
  - 利用停止させる暗号アルゴリズム
    - MD4、MD5、SHA-1、3DES, RC4 など
  - 推奨するアルゴリズムの対象
    - 認証暗号　AEAD(GCM, CCM), ChaCha20-Poly1305
    - 楕円曲線　Curve25519, Curve448
    - 署名　Edwards-Curve Digital Security Algorithm (EdDSA)
    - 鍵共有　ECDHE with Curve25519/Curve448(X25519/X448)

イメージ的に俯瞰すると・・・

CRYPTREC、IETF/IRTF、OSSの3領域で可視化



IETFが活発になっている現時点において実社会とのギャップが年々大きくなってきている

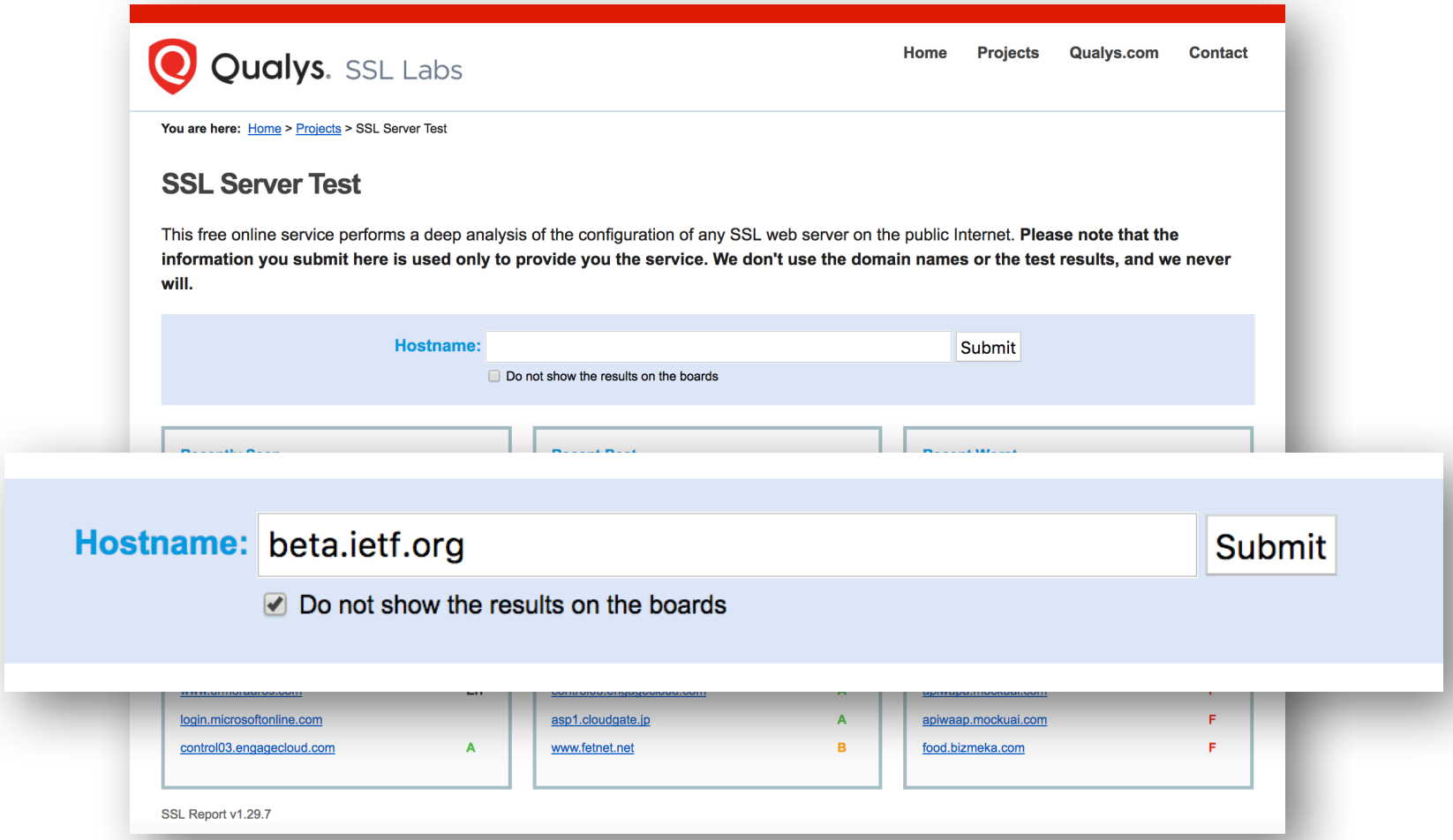
影響把握

- IETFでの標準化活動が、我々の身の回りにどの程度の影響を与えているのかを探る
  - 調査対象プロトコル：SSL/TLSプロトコル
  - 調査対象ソフトウェア：
    - ブラウザ：Chrome、Firefox、Edge
    - サーバ：OpenSSL
  - 調査手段：
    - Qualys SSL Labs
      - https://www.ssllabs.com/

参考情報：Qualys SSL Labs

追加

- ホスト名を入力することでSSL/TLSの設定を把握



参考情報：Qualys SSL Labs

追加

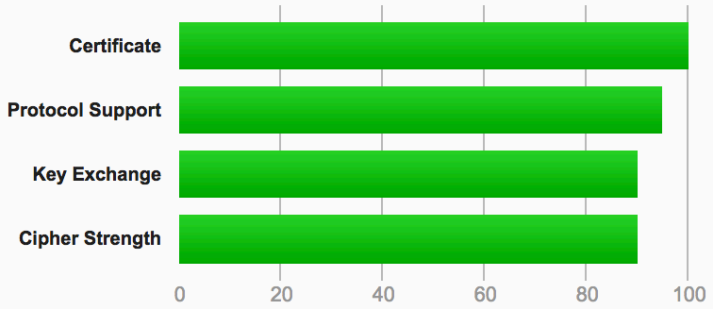
You are here: Home > Projects > SSL Server Test > beta.ietf.org > 104.20.1.85

SSL Report: beta.ietf.org (104.20.1.85)

Summary

Overall Rating

A+



Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

Experimental: This server supports TLS 1.3 (draft 18).

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. MORE INFO »

Certificate #1: RSA 2048 bits (SHA256withRSA)

Server Key and Certificate #1

Subject	*.ietf.org
	Fingerprint SHA256: e6a2d2a358a96964968c04f1644aedc435cc368a776403274dfc27921787b347

参考情報：Qualys SSL Labs

追加

Configuration

Protocols

TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

For TLS 1.3 tests, we currently support draft version 18.

Cipher Suites

 # TLS 1.3 (server has no preference)

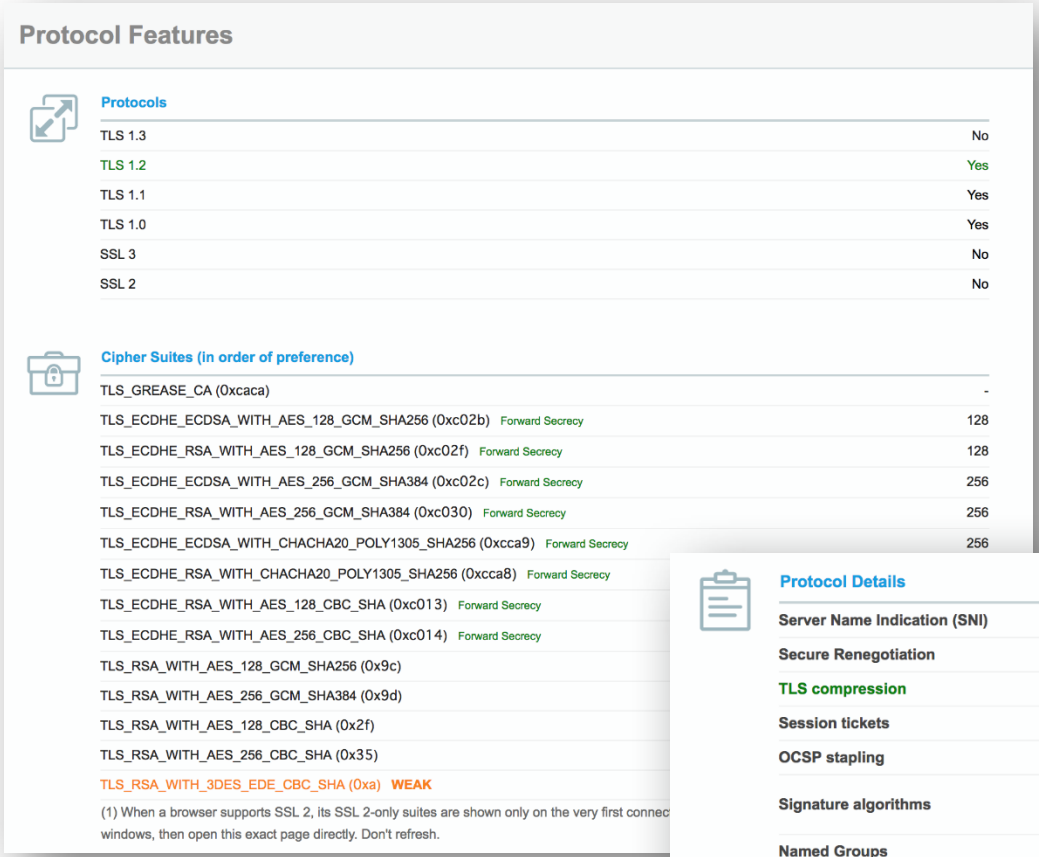
TLS_AES_128_GCM_SHA256 (0x1301)	ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_AES_256_GCM_SHA384 (0x1302)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303)	ECDH x25519 (eq. 3072 bits RSA) FS	256

 # TLS 1.2 (suites in server-preferred order)

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	ECDH x25519 (eq. 3072 bits RSA) FS	128
OLD_TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)	ECDH x25519 (eq. 3072 bits RSA) FS	128

影響把握：ブラウザ（1/3）

- Chrome 62.0.3202.94



Protocol Features

Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_GREASE_CA (0xcaca)	-
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) Forward Secrecy	
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Forward Secrecy	
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Forward Secrecy	
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connec
windows, then open this exact page directly. Don't refresh.

Protocol Details

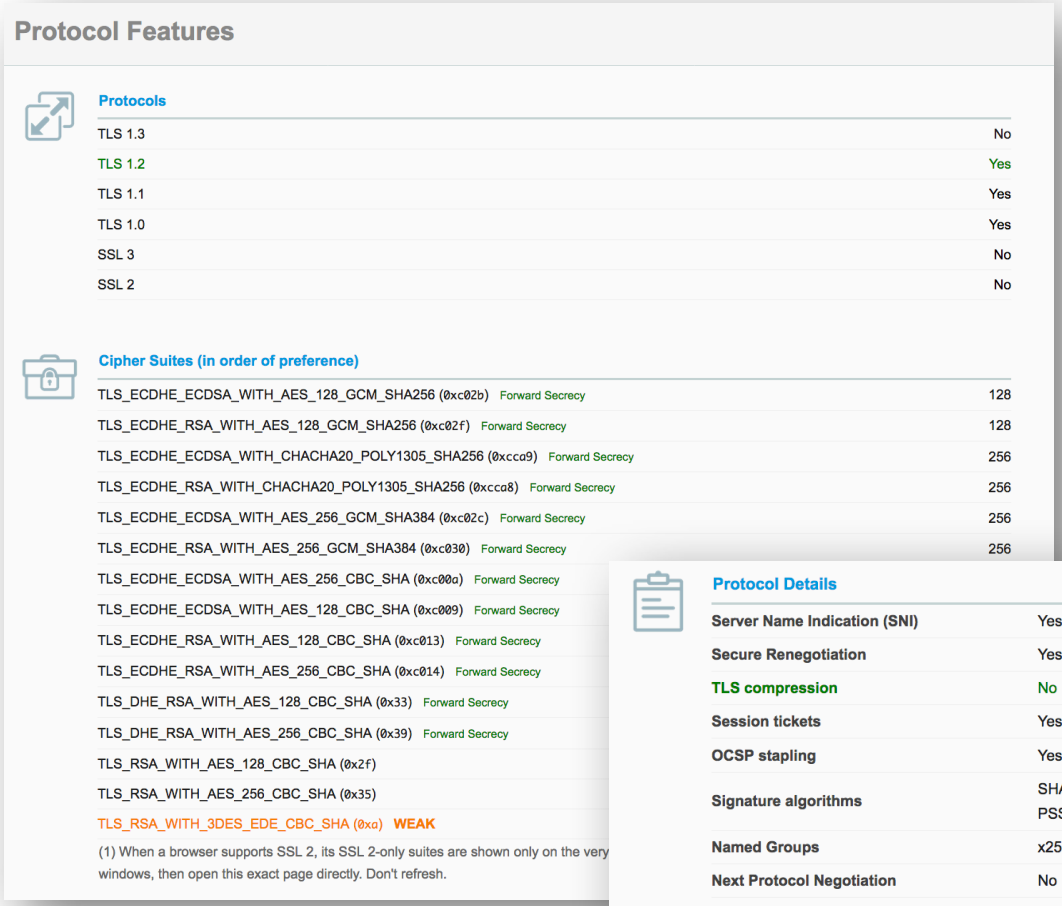
Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA256/ECDSA, RSA_PSS_SHA256, SHA256/RSA, SHA384/ECDSA, RSA_PSS_SHA384, SHA384/RSA, RSA_PSS_SHA512, SHA512/RSA, SHA1/RSA
Named Groups	tls_grease_aaaa, x25519, secp256r1, secp384r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

- ChaCha20-Poly1305対応
- AEADにほぼ対応
- ほぼECDHEに移行
- 楕円曲線 x25519対応

→ *ほぼ*IETFの推奨設定

影響把握：ブラウザ（2/3）

- Firefox 57.0 (64-bit)



Protocol Features

Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) Forward Secrecy	
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) Forward Secrecy	
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Forward Secrecy	
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Forward Secrecy	
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) Forward Secrecy	
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) Forward Secrecy	
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very
windows, then open this exact page directly. Don't refresh.

Protocol Details

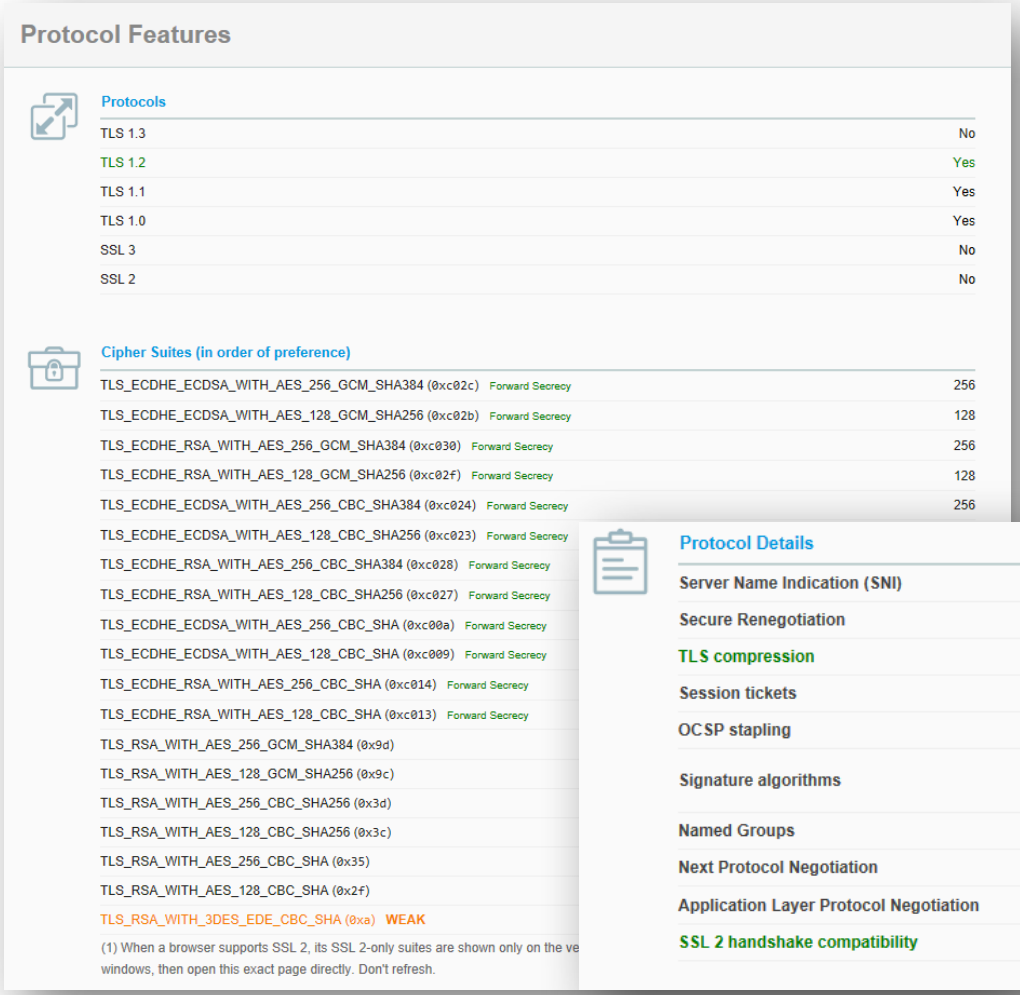
Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA256/ECDSA, SHA384/ECDSA, SHA512/ECDSA, RSA_PSS_SHA256, RSA_PSS_SHA384, RSA_PSS_SHA512, SHA256/RSA, SHA384/RSA, SHA512/RSA, SHA1/ECDSA, SHA1/RSA
Named Groups	x25519, secp256r1, secp384r1, secp521r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

- ChaCha20-Poly1305対応
- ほぼECDHE/DHEに移行
- AEADにほぼ対応
- 楕円曲線 x25519対応

→ IETFでの推奨状況

影響把握：ブラウザ（3/3）

- Microsoft Edge 40.15063.674.0



Protocol Features

Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023) Forward Secrecy	
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) Forward Secrecy	
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) Forward Secrecy	
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) Forward Secrecy	
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) Forward Secrecy	
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) Forward Secrecy	
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) Forward Secrecy	
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d)	
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c)	
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the ve
windows, then open this exact page directly. Don't refresh.

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA256/RSA, SHA384/RSA, SHA1/RSA, SHA256/ECDSA, SHA384/ECDSA, SHA1/ECDSA, SHA1/DSA, SHA512/RSA, SHA512/ECDSA
Named Groups	x25519, secp256r1, secp384r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

- ChaCha20-Poly1305未対応
- ほぼECDHEに移行
- AEADにほぼ対応
- 楕円曲線 x25519対応

→ IETFでの推奨状況

影響把握：サーバ（1/2）

- OpenSSL 1.1.0f

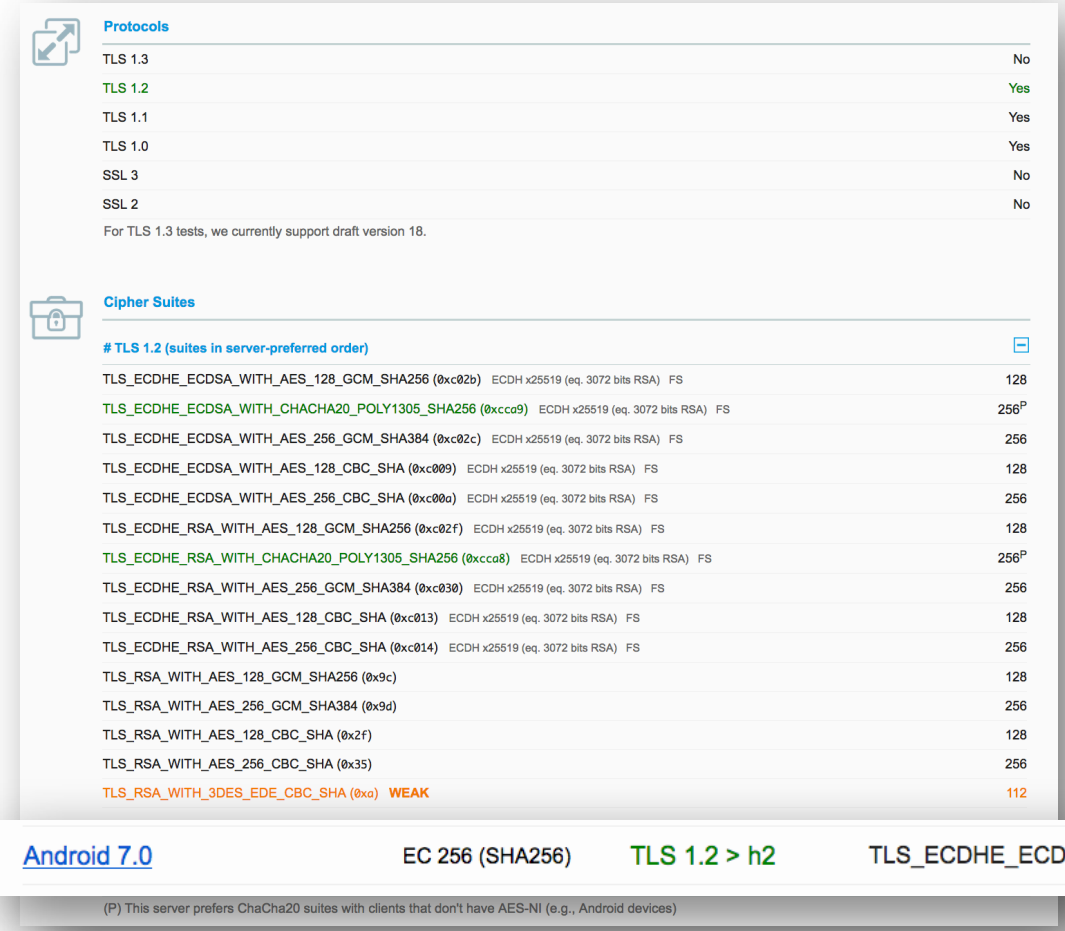
```
$ ./openssl ciphers -V
        0xC0,0x2C - ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH     Au
        0xC0,0x30 - ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH     Au=R
        0x00,0x9F - DHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=DH       Au=RSA
        0xCC,0xA9 - ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH     Au
        0xCC,0xA8 - ECDHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=ECDH     Au=R
        0xCC,0xAA - DHE-RSA-CHACHA20-POLY1305 TLSv1.2 Kx=DH       Au=RSA
        0xC0,0x2B - ECDHE-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au
        0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=R
        0x00,0x9E - DHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=DH       Au=RSA
        0xC0,0x24 - ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=ECD
        0xC0,0x28 - ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=RSA  En
        0x00,0x6B - DHE-RSA-AES256-SHA256   TLSv1.2 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA256
        0xC0,0x23 - ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=ECDSA Enc=AES(128)  Mac=SHA256
        0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA256
        0x00,0x67 - DHE-RSA-AES128-SHA256   TLSv1.2 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA256
        0xC0,0x0A - ECDHE-ECDSA-AES256-SHA  TLSv1 Kx=ECDH     Au=ECDSA Enc=AES(256)  Mac=SHA1
        0xC0,0x14 - ECDHE-RSA-AES256-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(256)  Mac=SHA1
        0x00,0x39 - DHE-RSA-AES256-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
        0xC0,0x09 - ECDHE-ECDSA-AES128-SHA  TLSv1 Kx=ECDH     Au=ECDSA Enc=AES(128)  Mac=SHA1
        0xC0,0x13 - ECDHE-RSA-AES128-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA1
        0x00,0x33 - DHE-RSA-AES128-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
        0x00,0xAD - RSA-PSK-AES256-GCM-SHA384 TLSv1.2 Kx=RSAPSK   Au=RSA  Enc=AESGCM(256) Mac=AEAD
        0x00,0xAB - DHE-PSK-AES256-GCM-SHA384 TLSv1.2 Kx=DHEPSK   Au=PSK  Enc=AESGCM(256) Mac=AEAD
        0xCC,0xAE - RSA-PSK-CHACHA20-POLY1305 TLSv1.2 Kx=RSAPSK   Au=RSA  Enc=CHACHA20/POLY1305(256) Mac=AEAD
        0xCC,0xAD - DHE-PSK-CHACHA20-POLY1305 TLSv1.2 Kx=DHEPSK   Au=PSK  Enc=CHACHA20/POLY1305(256) Mac=AEAD
        0xCC,0xAC - ECDHE-PSK-CHACHA20-POLY1305 TLSv1.2 Kx=ECDHEPSK Au=PSK  Enc=CHACHA20/POLY1305(256) Mac=AEAD
        0x00,0x9D - AES256-GCM-SHA384       TLSv1.2 Kx=RSA      Au=RSA  Enc=AESGCM(256) Mac=AEAD
        0x00,0xA9 - PSK-AES256-GCM-SHA384   TLSv1.2 Kx=PSK      Au=PSK  Enc=AESGCM(256) Mac=AEAD
        0xCC,0xAB - PSK-CHACHA20-POLY1305   TLSv1.2 Kx=PSK      Au=PSK  Enc=CHACHA20/POLY1305(256) Mac=AEAD
```

- ChaCha20-Poly1305対応
- ほぼECDHE/DHEに移行
- AEADにほぼ対応

→ IETFでの推奨状況

影響把握：サーバ（2/2） [参考情報, Google]

- 世界で一番 前のめりなサーバ設定



Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

For TLS 1.3 tests, we currently support draft version 18.

Cipher Suites

 # TLS 1.2 (suites in server-preferred order)

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9) ECDH x25519 (eq. 3072 bits RSA) FS	256P
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) ECDH x25519 (eq. 3072 bits RSA) FS	256P
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	112

Android 7.0	EC 256 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 ECDH x25519 FS

(P) This server prefers ChaCha20 suites with clients that don't have AES-NI (e.g., Android devices)

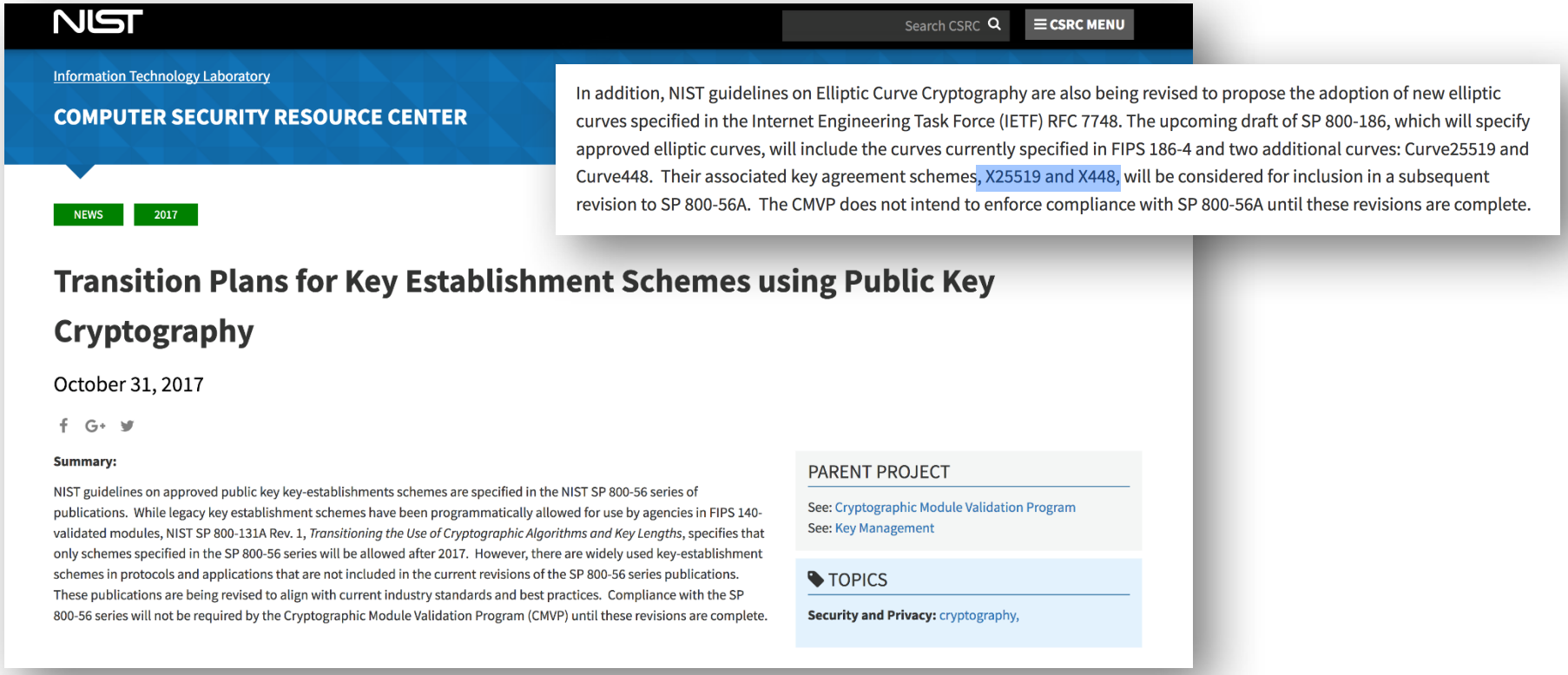
- Chromeと準拠
- ChaCha20-Poly1305対応
- ほぼECDHE/DHEに移行
- AEADにほぼ対応
- 楕円曲線 x25519対応

→ IETFでの推奨状況

※ Android7ではChaCha20を優先するサーバ設定

参考情報：NISTの動向

- IRTF cfrg で検討していたCurve25519およびCurve448の楕円曲線がFIPS186-4に追加へ！



NIST

Search CSRC

CSRC MENU

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

NEWS 2017

Transition Plans for Key Establishment Schemes using Public Key Cryptography

October 31, 2017

Summary:

NIST guidelines on approved public key key-establishments schemes are specified in the NIST SP 800-56 series of publications. While legacy key establishment schemes have been programmatically allowed for use by agencies in FIPS 140-validated modules, NIST SP 800-131A Rev. 1, *Transitioning the Use of Cryptographic Algorithms and Key Lengths*, specifies that only schemes specified in the SP 800-56 series will be allowed after 2017. However, there are widely used key-establishment schemes in protocols and applications that are not included in the current revisions of the SP 800-56 series publications. These publications are being revised to align with current industry standards and best practices. Compliance with the SP 800-56 series will not be required by the Cryptographic Module Validation Program (CMVP) until these revisions are complete.

PARENT PROJECT

See: Cryptographic Module Validation Program
See: Key Management

TOPICS

Security and Privacy: cryptography,

In addition, NIST guidelines on Elliptic Curve Cryptography are also being revised to propose the adoption of new elliptic curves specified in the Internet Engineering Task Force (IETF) RFC 7748. The upcoming draft of SP 800-186, which will specify approved elliptic curves, will include the curves currently specified in FIPS 186-4 and two additional curves: Curve25519 and Curve448. Their associated key agreement schemes, X25519 and X448, will be considered for inclusion in a subsequent revision to SP 800-56A. The CMVP does not intend to enforce compliance with SP 800-56A until these revisions are complete.

https://csrc.nist.gov/News/2017/Transition-Plans-for-Key-Establishment-Schemes

CRYPTRECとIETFの比較に関するまとめ

- IETFの活動が活発することによる影響
  - 実際に利用されるソフトウェアと実装されている暗号とCRYPTREC暗号リストにギャップが存在
- 自分自身が意図していない暗号利用
  - CRYPTREC暗号リストで推奨されていない暗号アルゴリズムを知らないうちに利用する懸念

- 参考情報
  - NISTはIETF/IRTFの暗号技術動向に追従する流れ

通信プロトコルにおける今後の展望

IETFにおける通信プロトコルに関する今後の展望を示す

- After IETF88で発生した潮流は継続しそう
  - IETFではIoTで利用するプロトコルを検討するWGが7程度あったりと力が入っている
  - 影響範囲は拡大する方向
- OSSコミュニティとも近い関係であるため、「標準化←→実装」の連携がシームレス
  - RFCで発行されるInternet Draftの段階から、実装のための検討が始まり、RFC発行と同時期に有効になる

「フォーラム標準」と「OSS活動」の重要性が増加

本日のアジェンダ

- 準備
  - IETFとは
- 本題
  - IETFと暗号アルゴリズムの関係
  - IETFでの暗号が使われている通信プロトコル
  - 通信プロトコルの現状
    - 具体例：SSL/TLS
    - 暗号技術の側面からCRYPTRECとIETFを比較
  - 今後の展望
- 我々はどうすべきか
  - 暗号利用の力学が変わってきているのでは？
  - インターネットでの暗号技術の利用に追従するには・・
  - CRYPTRECに期待すること
- まとめ

暗号利用の力学が変わってきているのでは？

- 全てのモノがネットワークに繋がる時代へ

- 構成要素としての暗号技術が素晴らしくても繋がらなければ、世界で利用されない
  - 言い換えれば、通信プロトコルで世界中で広く使われるためには「繋げるための努力」が重要
    - 「オープンな仕様」＆「動作する実装」の重要性が増大

この世界を非常に上手に泳いでるのは **Google**

インターネットでの暗号技術の利用に追従するには・・

- IETFでの新しい暗号技術を採用したRFCが発行されると、OSSコミュニティで実装し始めるため、新しい暗号アルゴリズムが利用できるライブラリ（実行環境）が世界中に広がる
  - その速度が年々早くなってきている（ように感じる）

今まで以上に「デファクト標準」への関心を増す必要があると考える

CRYPTRECに期待すること

すでに対応されているかも知れませんが・・・

- CRYPTREC暗号リストに準拠した利用環境を示す
  - 利用者視点での貢献
    - 例えば、お墨付きのソフトウェアや設定方法
- 情報収集が手薄になっているフォーラム標準化団体における技術的な動向を捉え、いち早く評価する対象を選定する
  - 実社会とのギャップを最小化するための活動
- ”待ち”から”攻め”に転じてデジュール標準と同様にフォーラム標準にも貢献する
  - 世界でのプレゼンス向上に向けて

我々、日本としてできること

- インターネットはギークのためのモノではく
  日常的に利用する生活基盤へ
  - 安心・安全に利用できることが大前提

  - 暗号技術の観点からIETFでの貢献を見ると、日本は「タダ乗り」している状況ではないか？
    - 要因：研究者は研究成果に繋がらなかったり、企業として売上や利益に繋がらない謎な活動として認識

日本の技術・研究レベルの高いので
彼らの研究成果をインターネットに積極的に
還元するというのはどうか？

※ 難しいのは承知で・・・

本日のアジェンダ

- 準備
  - IETFとは
- 本題
  - IETFと暗号アルゴリズムの関係
  - IETFでの暗号が使われている通信プロトコル
  - 通信プロトコルの現状と今後の展望
    - 具体例：SSL/TLS
- 我々はどうすべきか
  - 暗号利用の力学が変わってきているのでは？
  - インターネットでの暗号技術の利用に追従するには・・
  - CRYPTRECに期待すること
- まとめ

まとめ

- IETFでの実績が、暗号技術の観点からも我々の生活に与える影響が、より大きくなってきている
  - 例：cfrgでの選定した暗号アルゴリズム
    - ChaCha20-Poly1305、Curve25519 など
- CRYPTRECも設立時に想定していたスコープや速度感を現状に適用させるために変化しているが、インターネットを取り巻く環境は更に早いのでもう少しのテコ入れを期待
  - 暗号技術は「国家の安全保障」にも関連しますし・・・

何か気になることなどあれば・・・

- E-mail
  - kanno@lepidum.co.jp
- SNS
  - Twitter(satorukanno)
  - Facebook(satoru.kanno)

お気軽にご連絡ください！