

Internet Week 2015

S8 SSL/TLSはようになっていくのか(11/18水 9:30-)

SSL/TLSをめぐる最新動向

(講演資料)

2015年11月18日(水) 9:40-10:30 (50分)

於：富士ソフト アキバプラザ 6F Room3



富士ゼロックス株式会社
Fuji Xerox CERT 漆嵐 賢二

本文中の登録商標および商標はそれぞれの所有者に帰属します。

自己紹介: 漆 嵐 賢二(うるしま), CISSP

• 経歴

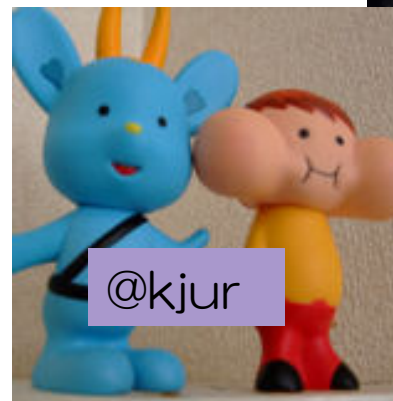
- 富士ゼロックス(2010~)
- エントラストジャパン(2005~2010)
- セコム(1988~2005)

• 興味:

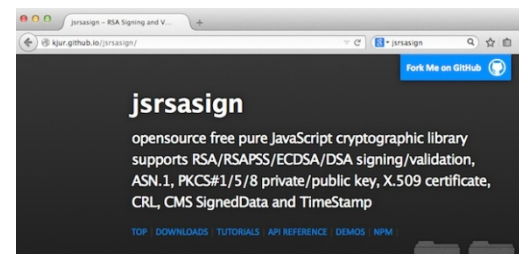
PKI, TLS, 電子署名, SSO, 認証, 暗号, CSIRT, 脆弱性検査, フォレンジック, スマホ, プログラミング, ビットコイン

• 別名

- 証明書ハンター
- (TLS)暗号スイートウォッチャー
- 委員、標準化、認定基準、実証実験、普及啓蒙
 - JNSA, CRYPTREC, 日本データ通信協会
 - IPAセキュキャン講師
 - IBECOM, PKI-J, 欧州ETSI
 - PKI, TLS, 長期署名, タイムスタンプ



ブログ: 自堕落な技術者の日記



jsrsasign - JavaScript 実装暗号ライブラリ

本日のアジェンダ

- SSL/TLSのおさらい
- 2015年のSSL/TLS関連に関連した話題
- 特に、Let's Encrypt！無料で簡単な証明書

SSL/TLSのおさらい



アマゾンの購入確認画面

出典：アマゾン(www.amazon.co.jp)

暗号化されているか

カード番号

氏名、住所、電話番号

何を買ったか？

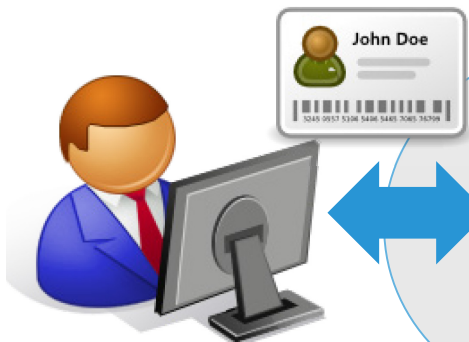
購入個数

The screenshot shows the Amazon Japan checkout page. The browser address bar shows a URL starting with 'https://www.amazon.co.jp'. The page title is '注文の確定 - Amazon.co.jp'. The main heading is '注文内容を確認・変更する'. Below this, there are several sections:

- お届け先住所 変更:** 漆島賢二, 東京都, 電話番号: [redacted]. A red box highlights this section.
- 支払い方法 変更:** VISA 下4桁 [redacted]. A red box highlights this section.
- Amazonギフト券・Amazonショッピングカードまたはクーポン:** コードを入力, 適用.
- お急ぎ便無料:** 今回のご注文から適用可能: 漆島賢二さん、右下の配送方法より「Amazonプライム無料体験登録で、お急ぎ便が無料に」を選択すれば、このご注文を無料のお急ぎ便でお届けします。
- お届け予定日:** 2015年11月2日.
- 商品:** イミテーション・ゲーム/エニグマと天才数学者の秘密 コレクターズ・エディション [初回限定生産]アウタースリーブ付 [Blu-ray] ベネディクト・カンバーバッチ. A red box highlights the product name and price (¥ 3,872). Below it, the quantity is '数量: 1 変更', also highlighted with a red box.
- 配送方法:** amazonプライム (本日 2015/11/1 日曜日 に お届けします), 通常配送 (当日お急ぎ便 本日 2015/11/1 日曜日 に お届けします), お届け日時指定便.

HTTPS暗号通信ってなんているの？

例えばアマゾンでお買い物



クレジットカード番号
住所・氏名
秘密にしたい買い物



SSL/TLSが守る



今のネットに必須の機能

「カード番号、住所、氏名、買い物の内容」を途中で見られたくない

ニセのアマゾンサイトに「カード番号、住所」なんかを送りたくない。



途中で「届け先住所」を書き換えて商品を騙し取られたくない。

SSL/TLSの3つの機能



「カード番号、住所、氏名、買い物の内容」を途中で見られたくない

ニセのアマゾンサイトに「カード番号、住所」なんかを送りたくない。

途中で「届け先住所」を書き換えて商品を騙しとられたくない。

機密性

暗号通信により通信相手以外に通信内容を盗み見(盗聴)されないようにする

覗き見(盗聴)防止

共通鍵暗号を使う

相手認証

証明書(PKI)などを使い通信相手が正しい相手であるか認証する

なりすまし防止

PKI(公開鍵暗号)、パスワード、Kerberos認証を使う

完全性

通信の途中でデータが書き換え(改ざん)されないよう、改ざん検知できる

改ざん防止

MAC(メッセージ認証コード)を使う

暗号スイート (Cipher Suite) とは

標準 (RFC) では
300種以上規定されている

ClientHello, ServerHelloでウェブブラウザとウェブ
サーバーが合意する暗号アルゴリズムのセット

ClientHello

----- TLS_RSA_WITH_RC4_128_MD5 ----->
----- TLS_RSA_WITH_DES_CBC_SHA ----->
----- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ----->
をサポートしてますけど、どうでしょうか?



ServerHello

<----- では、これをお願いします -----> 通信暗号強度が決まる
----- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ----- のでサーバー側では
注意が必要

(例)

TLS_RSA_WITH_AES128_CBC_SHA 値 0010

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 値 c02f

鍵交換と
(公開鍵暗号を使った)
認証のアルゴリズム

データの
共通鍵暗号の
アルゴリズム

メッセージ認証
(MAC)のアルゴリズム
※ハッシュ関数SHA1でなく
MAC関数 HmacSHA1

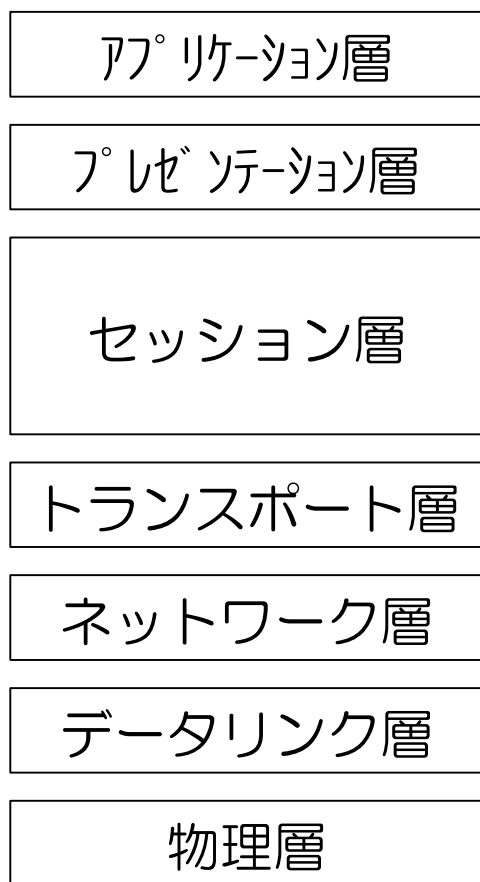
相手認証

データ暗号化

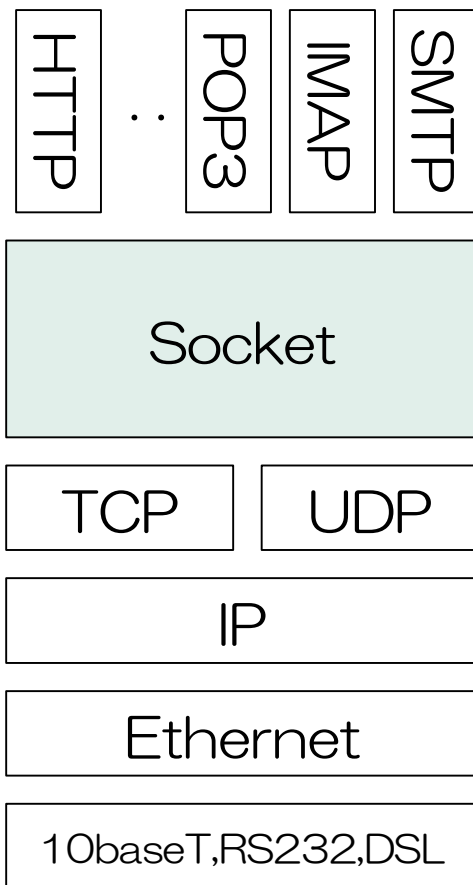
改ざん防止

SSL/TLSの特徴(HTTPでも何でも乗る)

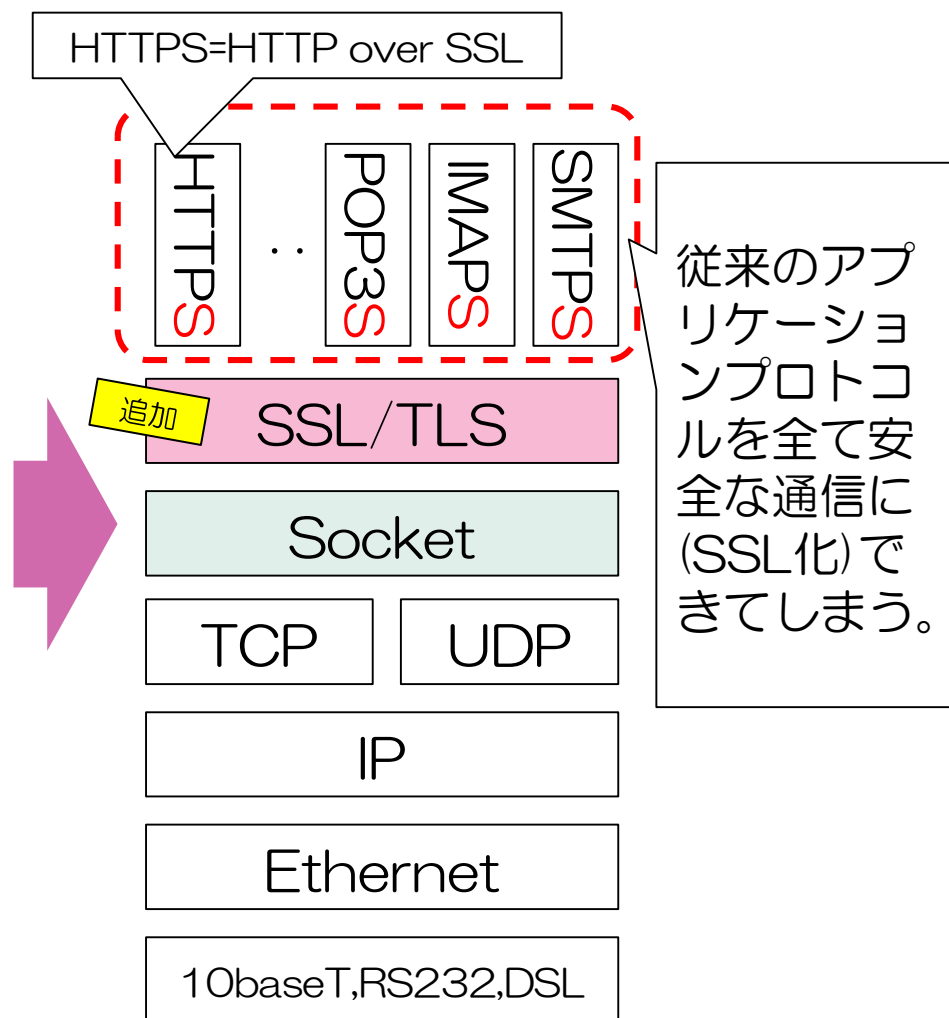
OSI参照モデル



従来通信の例



SSL化(※1)



※1: 他にLDAPs, FTPS, TELNETSとか

今年のSSL/TLSのトピック 脆弱性問題など



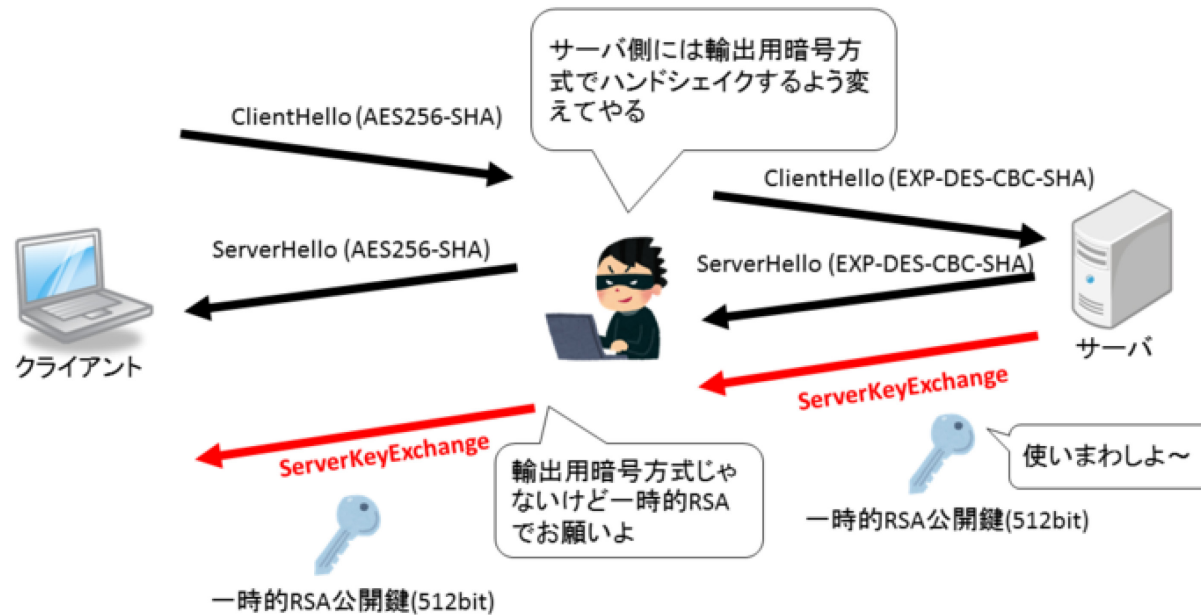
2015年のSSL/TLSのトピック/脆弱性問題

- ① 3月 FREAK脆弱性 (RSA輸出グレードの鍵の使用)
- ② 3月 Microsoft live.fiへの不適切な発行
- ③ 5月 SHA2証明書数がSHA1を抜く
- ④ 5月 IPA SSL/TLS設定ガイドライン公開
- ⑤ 6月 Logjam脆弱性 (弱いDH(E)鍵交換)
- ⑥ 7月 OpenSSL Alt Chain脆弱性 (パス長計算誤り)
- ⑦ 7月 RC4NOMORE (RC4が現実的に危殆化)
- ⑧ 8月 TLSのKCI攻撃 (fixed_(EC)DH鍵交換)
- ⑨ 9月 Thawteが勝手に不正google証明書発行
- ⑩ 11月 Let's Encrypt 無料自動証明書発行サービス開始

次点：Certificate Transparencyの普及

① 3月 FREAK攻撃(弱い輸出グレードのRSA鍵)

- クライアントが強い暗号スイートを要求しても、中間者がサーバーとは弱い輸出グレード(EXP)の暗号スイートを使わせてしまう
- EXP-RSAではサーバー起動後、常時同じRSA 512bitを使用する実装が多い
- RSA512は、Amazon EC2だと7時間で解けた



FREAK攻撃 その2

(出典：詳細解説) ぼちぼち日記：華麗なる因数分解:FREAK攻撃の仕組み <http://d.hatena.ne.jp/jovi0608/20150304/1425461359>

② 3月 Windows Live証明書の不適切な発行

- フィンランドの技術者が、フィンランドのWindows Liveで “hostmaster@live.fi” というドメイン管理者に見えるアカウントが取れてしまった。
- そのメールアドレスで試しに、“www.live.fi” ドメインのDV証明書をComodoから取ろうとしたら取れてしまった。
- これは、中間者攻撃にも使える問題のある証明書なので、Microsoftはこの証明書をブラックリストに入れた。
- CABF Baseline Profileでは、DVの確認方法として
 - a) ドメインの管理者のメールアドレスからの申請
 - b) ドメインのWhoisのメールアドレスからの申請
 - c) ドメインのウェブサーバーの管理ができることの証明(指定された場所に指定コンテンツが置ける事)

のいずれかであればよく、今回は a) の方法で、認証局としては問題ないが、DVには課題があることが露呈した。

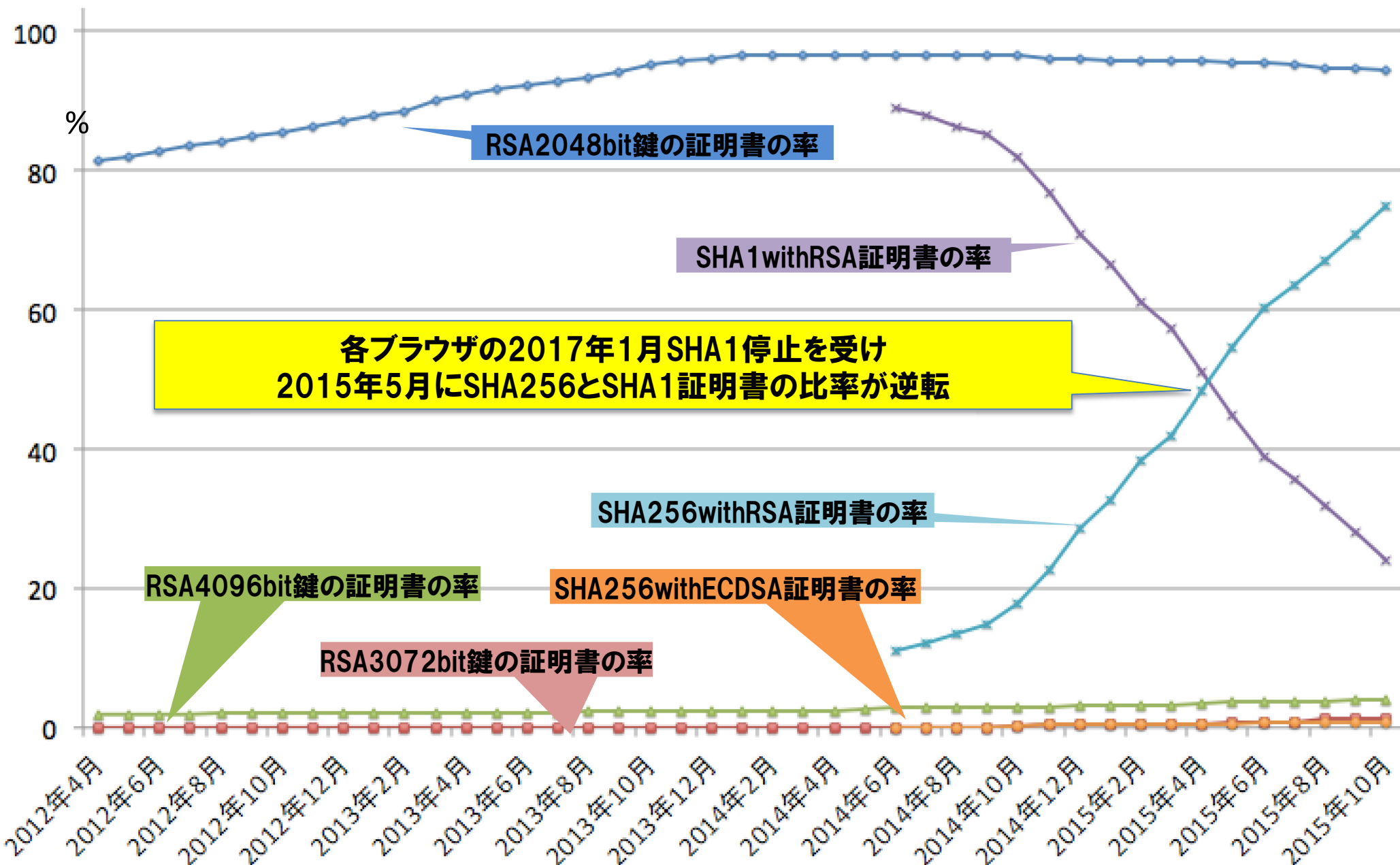


The screenshot shows a web browser displaying an article on the Ars Technica website. The URL in the address bar is <http://arstechnica.com/security/2015/03/man-who-obtained-windows-live-cert-said-his-warnings-went-unanswered/>. The article title is "Man who obtained Windows Live cert said his warnings went unanswered". The byline is "by Dan Goodin - Mar 17, 2015 11:23pm JST". The article text states: "A Finnish man who obtained an improperly issued HTTPS certificate for a Windows Live address said he warned both Microsoft and Finland authorities of the hole that made the security lapse possible but got no response, according to a published news article." There is a "FURTHER READING" section with a link to "BOGUS SSL CERTIFICATE FOR WINDOWS LIVE COULD ALLOW MAN-IN-THE-MIDDLE HACKS" and a small image of a banknote.

出典：

<http://arstechnica.com/security/2015/03/man-who-obtained-windows-live-cert-said-his-warnings-went-unanswered/>





















③ 5月 SHA2証明書がSHA1を抜く



③ 5月 SHA2証明書がSHA1を抜く

Microsoft製品、Google ChromeのSHA1証明書からの移行

- Windowsルート証明書プログラムのルートCA配下は2016年1月1日以降、SHA1証明書を発行できない。
- Windows製品では有効期限が2017年1月1日以降の証明書を受理しないためエラーとなる。
- Google ChromeはSHA1証明書の有効期限により、2015年1月以降のリリース版より下表の警告表示を開始し、2017年1月以降は受理しない。

Google Chrome		SHA1証明書の有効期限			
Chromeバージョン	安定版リリース日	2015.12.31 まで	2016.05.3 1	2016.12.3 1	2017.01 以降
38	2014.10.08	 https://	 https://	 https://	 https://
39	2014.11.18	 https://	 https://	 https:// ☆	 https:// ☆
40	2015.01.21	 https://	 https:// ☆	 https:// ☆	 https:// ☆
42	2015.04中旬	 https:// ☆	 https:// ☆	 https:// ☆	 https:// ☆
2017.01直後	2017.01中旬	 https:// ☆	 https:// ☆	 https:// ☆	 https:// ☆

記号：☆：Googleのポリシーにより、★：証明書期限切れにより

④ CRYPTREC/IPAのSSL/TLS暗号設定ガイド

The cover features a dark green header with the title "SSL/TLS 暗号設定ガイドライン" in white. Below it is a subtitle "～安全なウェブサイトのために(暗号設定対策編)～". A yellow box on the left contains the text "IPA版". A cartoon character with a mustache, wearing a white lab coat and a tie, is pointing upwards. At the bottom, the logos for "作成 CRYPTREC" and "発行 IPA" are displayed, along with the full name of the IPA Security Center.

http://www.ipa.go.jp/security/vuln/ssl_crypt_config.html

The screenshot shows a search bar with "IPA SSL ガイド" entered and a "検索" button. Below the search bar, the title "SSL暗号設定ガイドライン" is displayed. A yellow box on the left contains the text "CRYPTREC版". A tilted box indicates the release date "2015年5月に公開". The date "平成27年5月" is also shown. At the bottom, the logos for "独立行政法人 情報処理推進機構" and "国立研究開発法人 情報通信研究機構" are visible.

http://www.cryptrec.go.jp/topics/cryptrec_20150522_oper_guideline_fy2014.html

④ CRYPTREC/IPAのSSL/TLS暗号設定ガイド (非公式) 設定ファイル自動生成ツール

https://kjur.github.io/jsrsasign/tool_httpscfg.html

- 基本、「お好みのガイド」と「サーバーの種類」を選ぶだけ。
- CRYPTREC/IPAガイドを含む、様々なガイドラインに準拠したHTTPS設定ファイルを自動生成します。
- 今は、Apache HTTP 2.2/2.4、nginx、lighttpdに対応しています。
- 証明書(PEM)を貼れば、Certificate Pinningの鍵ハッシュ計算も自動で行います。
- ガイド種類はCRYPTREC, NIST, Mozilla, Bulletproof他、OSデフォルトもサポート

HTTPS設定ファイル生成ツール 0.5(ベータ版)

各種ガイドラインに応じたApache、nginx、lighttpdなど主要なサーバーのHTTPS設定を自動生成します

TOP

簡易設定情報

ガイドラインの種類：
CRYPTREC/IPAガイド(2015.05)高セキュリティ型・高いセキュリティを求める医療・金融・政府機関等向け

サーバーソフトウェアの種類： Apache HTTP Server 2.4

a) Pinにマッチさせる証明書のPEM(SSLサーバー証明書や中間CA証明書等)

```
-----BEGIN CERTIFICATE-----
MIIBdTCCAR+gAwIBAgIBBTANBgkqhkiG9w0BAQUFADAaMQswCQYDVQQGEwJVUzEL
MAkGA1UECgwCYTEwHhcNMjMwNTA0MDM0MTQxWhcNMjMwNTA0MDM0MTQxWjAa
MQsw
```

b) Pinにマッチさせないバックアップの証明書もしくはPKCS#8公開鍵のPEM

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAA33TqgLR3eeUmDtHS89qF
3p4MP7Wfat2Zjj3LzLjjCGDvvr9cJNlNDiuKbo0DgUiT4ZdPwb0iMAFdcDz10xA
```

生成 Reset

生成された設定ファイルの一部

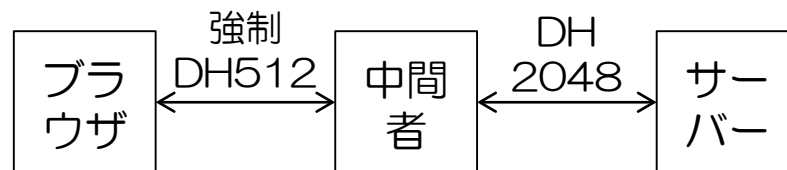
```
# sample ssl.conf for Apache 2.4
Listen 443 https
SSLPassPhraseDialog exec:/usr/libexec/httpd-ssl-pass-dialog
SSLSessionCache shmcb:/run/httpd/sslcache(512000)
SSLSessionCacheTimeout 300
SSLRandomSeed startup file:/dev/urandom 256
SSLRandomSeed connect builtin
SSLCryptoDevice builtin

<VirtualHost _default_:443>
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
```

⑤ Logjam(弱いDH(E) 鍵脆弱性)

Matthew Green教授のブログでの解説

- Logjam脆弱性は、鍵交換で輸出グレードのDH(E) 鍵交換(DHE_EXPORT、DH 512bit)に中間者がダウングレードすることにより通信を盗聴する脆弱性
- 利用者に気づかれないように、タイムアウトしないよう工夫するとしても、数秒～数分のうちに鍵を解読する必要あり
- DH鍵の解読は2ステップ。大量の計算パワーの必要とするDHパラメータ共通な事前計算と、鍵交換毎の計算
- 512bitなら、個人でもこの程度のクラスで1、2分で解読可能
- 1024bitなら、NSAの予算規模を考えれば実現可能性が高い。実際、SSHのDH鍵の解読プロジェクトがスノーデンの文書で明らかになっているので応用は容易
- DHでたった2つの素数が92%のApache/mod_sslで使われていたり、組み込み、ハードコード、OSのデフォルトで使っているケースも多く有名な鍵の事前計算の価値は十分ある



A Few Thoughts on Cryptographic Engineering

Some random thoughts about crypto. Notes from a course I teach. Pictures of my dachshunds.

Friday, May 22, 2015

Attack of the week: Logjam

In case you haven't heard, there's a new SSL/TLS vulnerability making the rounds. Nicknamed **Logjam**, the new attack is 'special' in that it may admit *complete decryption* or hijacking of any TLS connection you make to an improperly configured web or mail server. Worse, there's at least circumstantial evidence that similar (and more powerful) attacks might already be in the toolkit of some state-level attackers such as the NSA.



This work is the result of an unusual collaboration between a fantastic group of co-authors spread all around the world, including institutions such as the University of Michigan, INRIA Paris-Rocquencourt, INRIA Paris-Nancy, Microsoft Research, Johns Hopkins and the University Of Pennsylvania. It's rare to see this level of collaboration between groups with so many different areas of expertise, and I hope to see a lot more like it. (Disclosure: I am one of the authors.)

The absolute best way to understand the Logjam result is to read the technical research paper. This post is mainly aimed at people who want a slightly less technical form. For those with even shorter attention spans, here's the TL;DR:

It appears that the **Diffie-Hellman** protocol, as currently deployed in SSL/TLS, may be vulnerable to a serious downgrade attack that restores it to 1990s "export" levels of security, and offers a practical "break" of the TLS protocol against poorly configured servers. Even worse, extrapolation of the attack requirements -- combined with evidence from the Snowden documents -- provides some reason to speculate that a similar attack could be leveraged against protocols (including TLS, IPsec/IKE and SSH) using 768- and 1024-bit Diffie-Hellman.

About Me



Matthew Green

I'm a cryptographer and research professor at Johns Hopkins University. I've designed and analyzed cryptographic systems used in wireless networks, payment systems and digital content protection platforms. In my research I look at the various ways cryptography can be used to promote user privacy.

My website

[My twitter feed](#)
[Useful crypto resources](#)
[RSS](#)
[Bitcoin tipjar](#)
[Matasano challenges](#)

[Journal of Cryptographic Engineering \(not related to this blog\)](#)

[View my complete profile](#)

Popular Posts

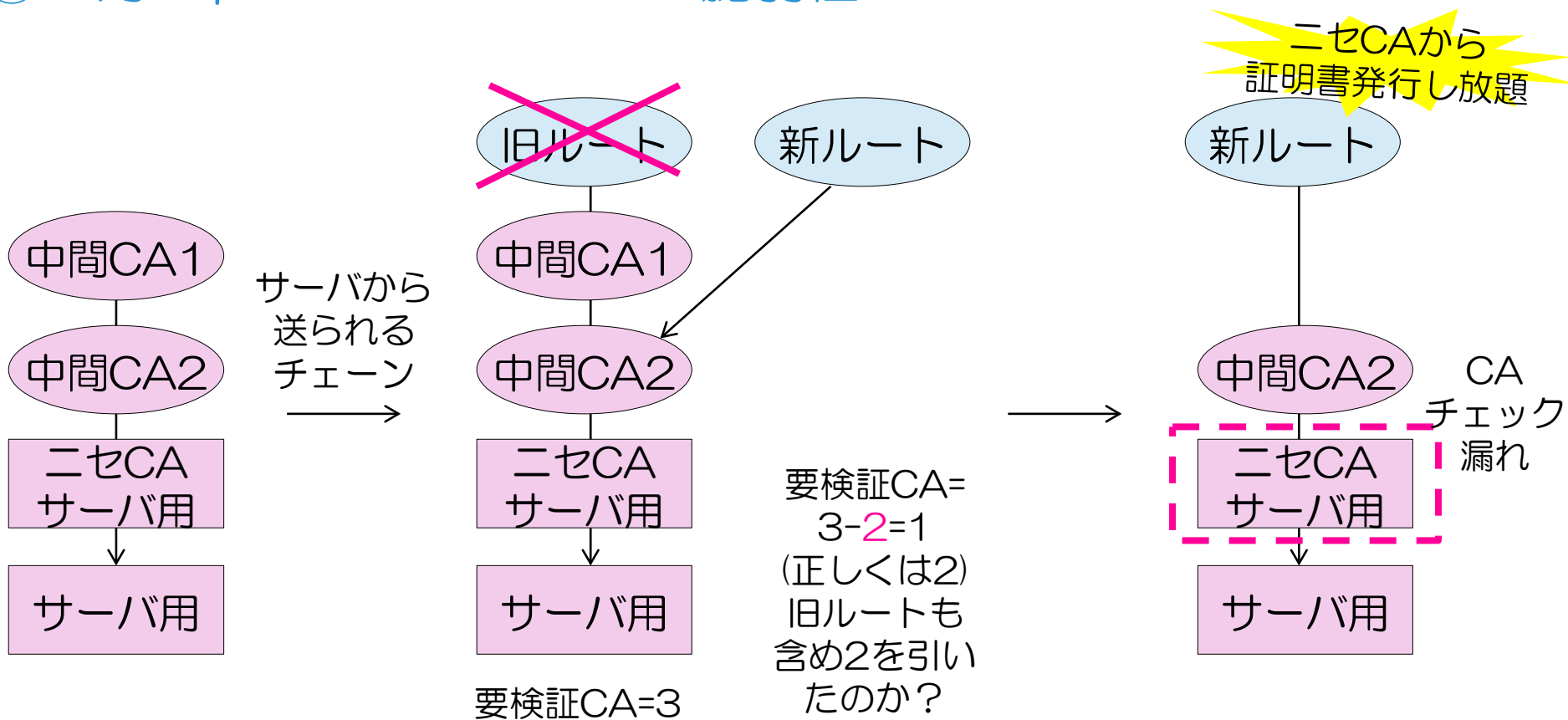


On the NSA
Let me tell you the story of my tiny brush with the biggest crypto story of the year. A few weeks ago I received a call from a reporter a...

出典:

<http://blog.cryptographyengineering.com/2015/05/attack-of-week-logjam.html>

⑥ 7月 OpenSSL Alt Chain脆弱性



- ルートCAの切替えがあっても、別のチェーンを探し検証するAlt Chainの機能が追加された
- チェーンの再計算の際、中間CA証明書数を数え間違い、問題に
- OpenSSLだけの問題

(参考詳細解説) <http://d.hatena.ne.jp/jovi0608/20150710/1436521488>

ぼちぼち日記: OpenSSLの脆弱性(CVE-2015-1793)によるAltチェーン証明書偽造の仕組み

⑦ 7月 RC4NOMORE脆弱性


- 2015年7月、さらに効率的にRC4暗号スイートの通信からクッキーを盗聴する方法が発表された
- 実機で52時間の盗聴により、暗号化されたセッションクッキーを収集し、93%と高確率のクッキー候補リストを生成、数分で正しいクッキーを発見できる。
- 2015年8月のUSENIX Securityカンファレンスで発表される
- 仕様上RFC 7465によりTLSv1.xの全てでRC4を使ってはならない事になった
- RC4の利用停止を考える時期にきた

RC4 NOMORE

Numerous Occurrence MOnitoring & Recovery Exploit
By *Mathy Vanhoef and Frank Piessens, iMinds-DistriNet, KU Leuven*

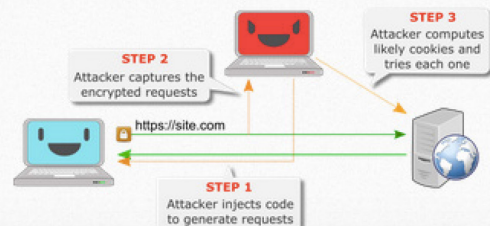
INTRODUCTION DEMO PAPER Q&A RESOURCES

INTRODUCTION

When you visit a website, and the browser's address bar contains a lock icon , the HTTPS protocol is used to protect your communication with this website (providing security and privacy). HTTPS supports several encryption techniques, one of them being the famous RC4 algorithm. At one point RC4 was used 50% of the time, with the latest estimate being 30%. Our RC4 NOMORE attack exposes weaknesses in this RC4 encryption algorithm. More precisely, in most situations where RC4 is used, these weaknesses can be used to reveal information which was previously thought to be safely encrypted.

In particular we show that an attacker can **decrypt web cookies**, which are normally protected by the HTTPS protocol. Websites use these cookies to identify users and authorize actions they perform. By obtaining the cookie of a victim, an attacker can log into a website as if he were the victim. This means the attacker can perform actions under the victim's name (e.g. post status updates and send messages), gain access to personal information (e.g. to emails and chat history), and so on.

The research behind the attack will be presented at [USENIX Security](#). Summarized, an attacker can decrypt a cookie within 75 hours. In contrast to previous attacks, this short execution time allows us to perform the attack in practice. When we tested the attack against real devices, it **took merely 52 hours** to successfully perform the attack. The attack consists of three steps:



```
graph LR; A[Attacker] -- "STEP 1: Attacker injects code to generate requests" --> B[Victim's Browser]; B -- "https://site.com" --> C[Server]; C -- "STEP 2: Attacker captures the encrypted requests" --> A; A -- "STEP 3: Attacker computes likely cookies and tries each one" --> C;
```

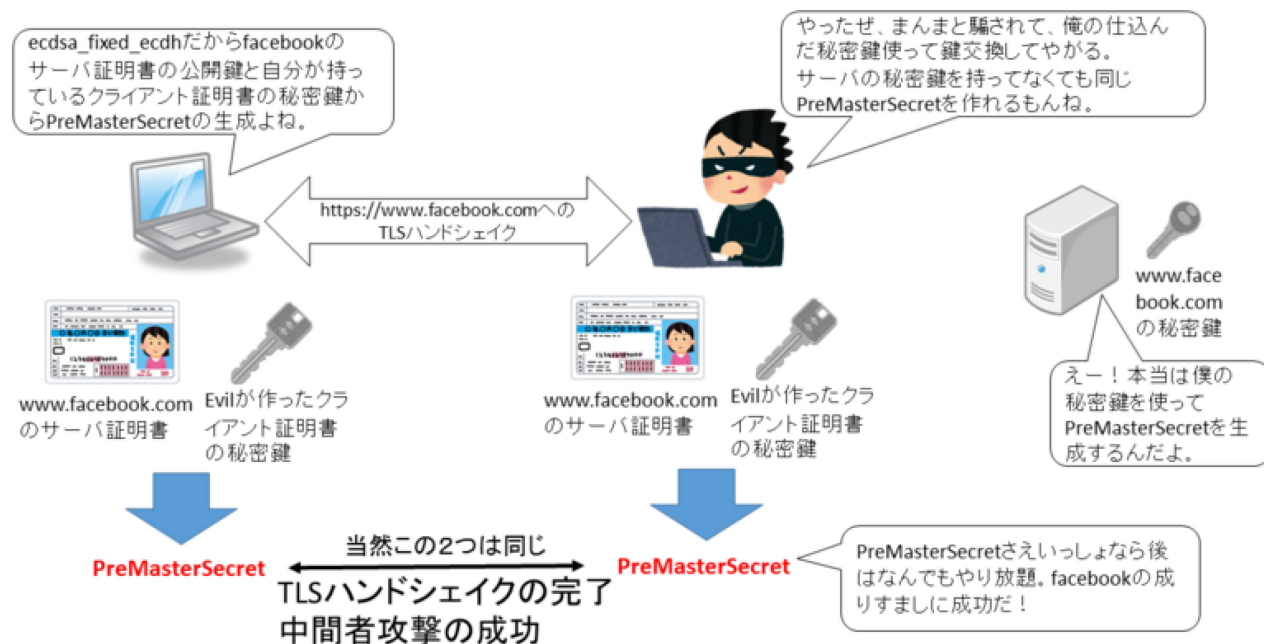
出典：RC4NOMORE

<http://www.rc4nomore.com/>

⑧ 8月 TLSのKCI中間者攻撃

- 疑った仕組みでサーバーとクライアントを騙し、中間攻撃者がPreMasterSecretを作って盗聴する。
- クライアント認証の場合には、サーバー証明書とクライアントの秘密鍵でPreMasterSecretが作れる。
- クライアントには攻撃者が作ったクライアント認証の鍵を仕込む。
- クライアントは古いSafariしか影響無し。サーバーは(EC)DSA証明書が必要

KCI攻撃ステップ4: 同一PreMasterSecretの生成



(出典) ぼちぼち日記：パンドラの箱？TLS鍵交換の落とし穴、KCI攻撃とは何か <http://d.hatena.ne.jp/jovi0608/20150821/1440117459>

⑨ 9月 Thawteが勝手に*.google.com証明書を発行

- Symantecの安価な証明書ブランドであるThawteがGoogleに断りなく勝手に、“*.google.com”のテスト用証明書を発行した。
- 運用上の問題がCTログを元に発見され、Googleは厳しく追求した。話し合いの下、証明書は失効された。
- 当該ドメインのワイルドカード証明書は、中東の政府による盗聴行為に使われるなど、危険性が高いものであり、Googleは、このような発行に対して非常にナーバスになっている。

The screenshot shows a BoingBoing article from September 19, 2015, by Cory Doctorow. The main headline is "Symantec caught issuing rogue Google.com certificates". The article features a banner for Symantec with the text "Symantec the First SSL choice." and "Lock down your IT Security". The article text discusses how Symantec was caught issuing rogue Google certificates, which are used to intercept communications. It mentions that Google created the Certificate Transparency initiative to address such issues. The article also includes a "POPULAR POSTS" section with links to other articles and a "FOLLOW BOING BOING" section with social media links.

出典：

<http://boingboing.net/2015/09/19/symantec-caught-issuing-rogue.html>

詳細解説(参考)：

自堕落な技術者の日記

http://blog.livedoor.jp/k_urushima/archives/1779810.html

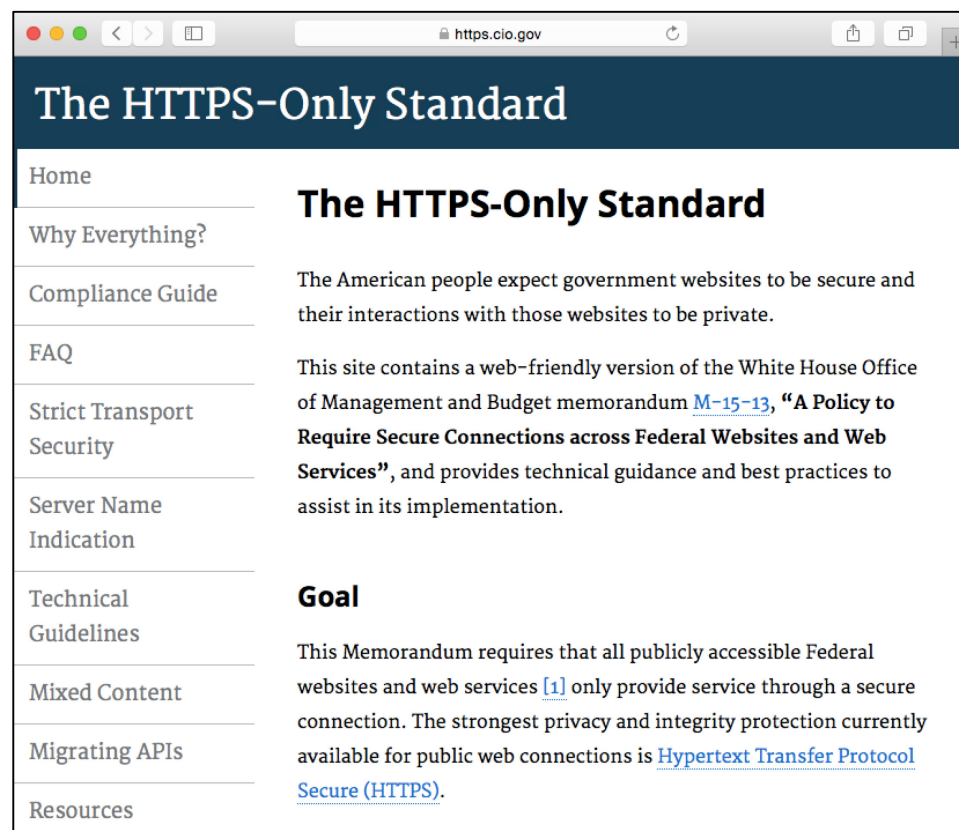
Let's Encrypt について

The screenshot shows the Let's Encrypt website homepage. At the top left is the Let's Encrypt logo, and at the top right is the text "LINUX FOUNDATION COLLABORATIVE PROJECTS" with navigation links for "Blog", "Technology", "Sponsors", "Support", and "About". The main content area features a large banner with a geometric background. The banner text reads: "Let's Encrypt is a new Certificate Authority: **It's free, automated, and open.** In Limited Beta". Below the banner, there are two columns: "FROM OUR BLOG" and "MAJOR SPONSORS". The blog section includes a post from Oct 29, 2015, titled "The CA's Role in Fighting Phishing and Malware" with a "Read more" link. The sponsors section displays logos for mozilla, Akamai, CISCO, E (Elastic), IdenTrust, and Internet Society.

出典：<https://letsencrypt.org/>

全てHTTPS化の流れ

- 昔は、ログイン画面と入力フォームだけHTTPS化しておけばよかった。
- 最近では、全ての通信をHTTPでなくHTTPSにすべきという動き
- 米政府HTTPS Only Standard



出典：https://https.cio.gov/

Let's Encrypt

- 全てHTTPSにするという世の中が進まないのは、SSLサーバー証明書が有料もしくは高価(数千円～数万円/年)だからでは？
- ISRG(インターネット研究の公益法人)がMozilla、Akamai、Ciscoなどをスポンサーとし、だれでも無料でSSL証明書を作る仕組みを作る

証明書の導入はとても面倒だった

- WindowsやFirefoxだと簡単に発行できるところもあるけれど、LinuxのウェブサーバーやSSLアクセラレーターに入れようとするするとOpenSSLコマンドを使ったり、結構面倒なステップを経て、証明書を発行してもらい、設定ファイルの作り方も結構面倒で、初めてなら戸惑うかもしれない。



**Let's
Encrypt**

- ✓ **簡単**
- ✓ **安全**
- ✓ **早い**
- ✓ **無料**
- ✓ **オープン**

Let's Encrypt の歴史

- 2014年頃 EFF, ミシガン大, Mozillaが無料証明書のプロジェクトを発足
- 2014年頃 ISRGを母体とすることした
- 2014年11月18日 Let's Encryptが2015年夏に無料証明書発行を告知
- 2015年1月28日 証明書簡易発行の Protokol ACME IETF I-D初版公開
- 2015年4月9日 ISRGとLinux Foundation協業発表
- 2015年6月16日 最初の証明書を6/27の週、一般向け9/14の週に計画発表
- 2015年8月7日 最初の証明書を9/7の週、一般向け11/16の週に計画延期
- 2015年9月14日 プライベートISRGルートCAから最初の証明書発行
- 2015年10月19日 パブリックCA(IdeTrust)との相互認証

- 2015年11月16日の週 一般向けサービス開始予定

計画は遅れがちなので、11月末に出ればいい方？

Let's Encrypt のSSL証明書はどんな人に向く？

Let's Encrypt のSSLサーバー証明書は、発行や証明書と鍵の設定が、かなりの部分自動化されたドメイン認証(DV)の証明書です。向き、不向きは以下の通りです。

向いている人

- Linux上でApache、Nginxの公開ウェブサーバーをHTTPS対応したい人
- 開発用サーバーをHTTPS化させたい人
- 証明書に会社や学校名が含まれなくても困らない人
- 証明書にお金を払いたくない人
- 旧来の証明書発行手続きを知らず、勉強したくない人
- コマンドライン一発で証明書発行してほしい人
- ウェブサーバーのHTTPS設定が面倒な人
- テスト用に無料でパブリックな証明書が欲しい人


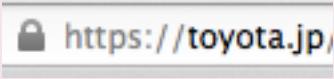
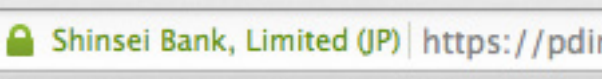
向かない人

- 上記以外のウェブサーバー、アプリケーションサーバー、SSLアクセラレーター、メールサーバー等に、SSLサーバー証明書を入れたい人
- 商用のサービスでHTTPS対応にする必要がある人
- 証明書に会社や学校名が入らないと困る人
- 証明書の代金支払に困っていない人

代表的な無料証明書との比較

	Lets Encrypt	WoSign	StartSSL
種類	DV	DV	DV
所要時間	数秒	数時間～数日	数日
長所	<ul style="list-style-type: none"> 専用コマンドで発行から設定まで簡単で速い ドメイン認証、サーバー設定まで自動 	<ul style="list-style-type: none"> ドメイン認証は、指定されたHTMLファイルをダウンロードし公開サーバーに設定するだけ 有効期間は1年 	<ul style="list-style-type: none"> 有効期間1年
短所	<ul style="list-style-type: none"> 標準対応でない環境だと戸惑う 有効期間が90日と短い 	<ul style="list-style-type: none"> 発行に数時間から数日かかる 	<ul style="list-style-type: none"> 管理者画面アクセスのために、クライアント証明書が必要で手順が煩雑 ドメイン認証は管理者メール認証 手間と時間がかかる

DV(ドメイン認証)証明書であることの注意点 / OV, EVとの違い

	DV(ドメイン認証)	OV(組織認証)	EV(拡張認証)
アドレス	OVと同じ 	DVと同じ 	緑アドレスバーに組織名が表示される 
ページ情報	組織名は表示されない Web サイトの識別情報 Web サイト: hikari-n.jp 運営者: 検証され信頼できる運営者情報はありません 認証局: COMODO CA Limited	組織名は表示されない Web サイトの識別情報 Web サイト: toyota.jp 運営者: 検証され信頼できる運営者情報はありません 認証局: Symantec Corporation	組織名が表示される Web サイトの識別情報 Web サイト: pdirecta08.shinseibank.c 運営者: Shinsei Bank, Limited 認証局: Entrust, Inc.
証明書 ビューアー	組織名は表示されない 発行対象 一般名称 (CN) hikari-n.jp 組織 (O) <証明書に記載されていません> 部門 (OU) Domain Control Validated	組織名は表示されない 発行対象 一般名称 (CN) toyota.jp 組織 (O) TOYOTA MOTOR CORPORATION 部門 (OU) e-TOYOTA DIV 03 シリアル番号 70:F0:A5:B1:5F:D8:A4:DF:A5:D7..	組織名が表示される 発行対象 一般名称 (CN) pdirecta08.shinseibank.com 組織 (O) Shinsei Bank, Limited 部門 (OU) ITDiv

企業のホームページでDV証明書を使うと、会社組織名が表示されないために、本当にその会社が運営しているサーバーか？フィッシングサイトでないか？利用者を不安にさせてしまう。

他サービスとの証明書を発行し サーバー設定するまでの所用時間の比較

	Lets Encrypt	WoSign(DV)	OV一般
時間	数秒～数分	数時間～数日	数日～数週間
手順	<ul style="list-style-type: none"> • コマンドに従い入力(数分) • 事前設定ファイルを使えば即時 	<ul style="list-style-type: none"> • アカウント作成(10分) • OpenSSL鍵ペア生成(数分) • 証明書発行要求生成(数分) • 必要情報入力送信(数分) • 証明書取得待ち(数時間～数日) • 証明書、HTTPS設定(1時間) 	<ul style="list-style-type: none"> • アカウント作成(10分) • OpenSSL鍵ペア生成(数分) • 証明書発行要求生成(数分) • 印鑑証明、会社登記の取得(数日) • 発行申請の社長印取得(数日) • 書類の送付(数日) • 電話による意思確認、実在確認 • 証明書取得待ち(数時間) • 証明書、HTTPS設定(1時間)

一般的なDV(ドメイン認証)の方法

3つのうちどれかでドメインを確認する

① 管理者メールアドレス

- {webmaster,hostmaster,root}@example.comなど管理者らしいメールアドレスからの申請か確認。
- Windows Live問題等、課題が発覚

② ウェブコンテンツ

- 認証局が指定した場所に、指定したコンテンツを置いてもらうことで、ドメインのウェブサーバーの管理権限があることを確認。
- 最近主流になりつつあり、Lets Encryptでも使っている

③ ドメインのWHOIS情報

- ドメインのWHOIS登録情報を確認し、DNS管理者からの申請であるかどうか確認する。
- 大きな企業ではドメイン管理者とウェブサーバー管理者は独立しており、この確認方法ではうまくいかないケースがある。

Let's Encrypt プレビュー版で
証明書を発行してみよう



Let's Encrypt の証明書発行、HTTPS設定の流れ

準備

- LinuxでHTTP公開ウェブサイトの構築
- Gitでletsencryptツールのインストール

実行

- ツールの実行(証明書取得と自動設定)

運用

- 秘密鍵が漏洩したらツールで失効
- 有効期限が切れそうならツールで更新

① 準備

Linux等で公開ウェブサーバーを立てる(DNS登録含む)

- ※Ubuntu, Debian, CentOS, Fedoraなど安定している。
FreeBSD, Mac OS Xもサポートしている
- ※ターミナルさえあればよく、サーバーでも可
- ※サーバー自動設定はApache、Nginxをサポート

Gitでletsencryptツールをダウンロード

- ※Pythonスクリプトで必要なパッケージは
自動追加インストールされる

```
% git clone http://github.com/letsencrypt/letsencrypt
```

- ※以降の画面は10月25日のプレビュー版時点での動作です。
最新版は変わっているかもしれません。本家や他のサイトでの
解説も古くてそのままでは動かないものが多いです。

② 実行(証明書の取得とサーバーの自動設定)

Apacheで公開サーバーが動いているとしてツール実行

```
% cd letsencrypt  
% sudo ./letsencrypt-auto
```

```
This is a PREVIEW RELEASE of a client application for the Let's  
Encrypt certificate authority and other services using the ACME  
protocol. The Let's Encrypt certificate authority is NOT YET ISSUING  
CERTIFICATES TO THE PUBLIC.
```

```
Until publicly-trusted certificates can be issued by Let's Encrypt,  
this software CANNOT OBTAIN A PUBLICLY-TRUSTED CERTIFICATE FOR YOUR  
WEB SERVER. You should only use this program if you are a developer  
interested in experimenting with the ACME protocol or in helping to  
improve this software. If you want to configure your web site with  
HTTPS in the meantime, please obtain a certificate from a different  
authority.
```

```
For updates on the status of Let's Encrypt, please visit the Let's  
Encrypt home page at https://letsencrypt.org/.
```

<Agree >

<Cancel >

プレビュー版であることの確認で<Agree>を選びリターン

② 実行(証明書の取得とサーバーの自動設定)

```
No names were found in your configuration files.  
You should specify ServerNames in your config files in order to  
allow for accurate installation of your certificate.  
If you do use the default vhost, you may specify the name manually.  
Would you like to continue?  
  
< Yes >           < No >
```

ApacheサーバーでFQDNが設定されていないので手入力の必要があるので<Yes>を選択しリターン

② 実行(証明書の取得とサーバーの自動設定)

Please enter in your domain name(s) (comma and/or space separated)

< OK > < Cancel >

サーバーのFQDN(=取得したい証明書のFQDN)を入力する

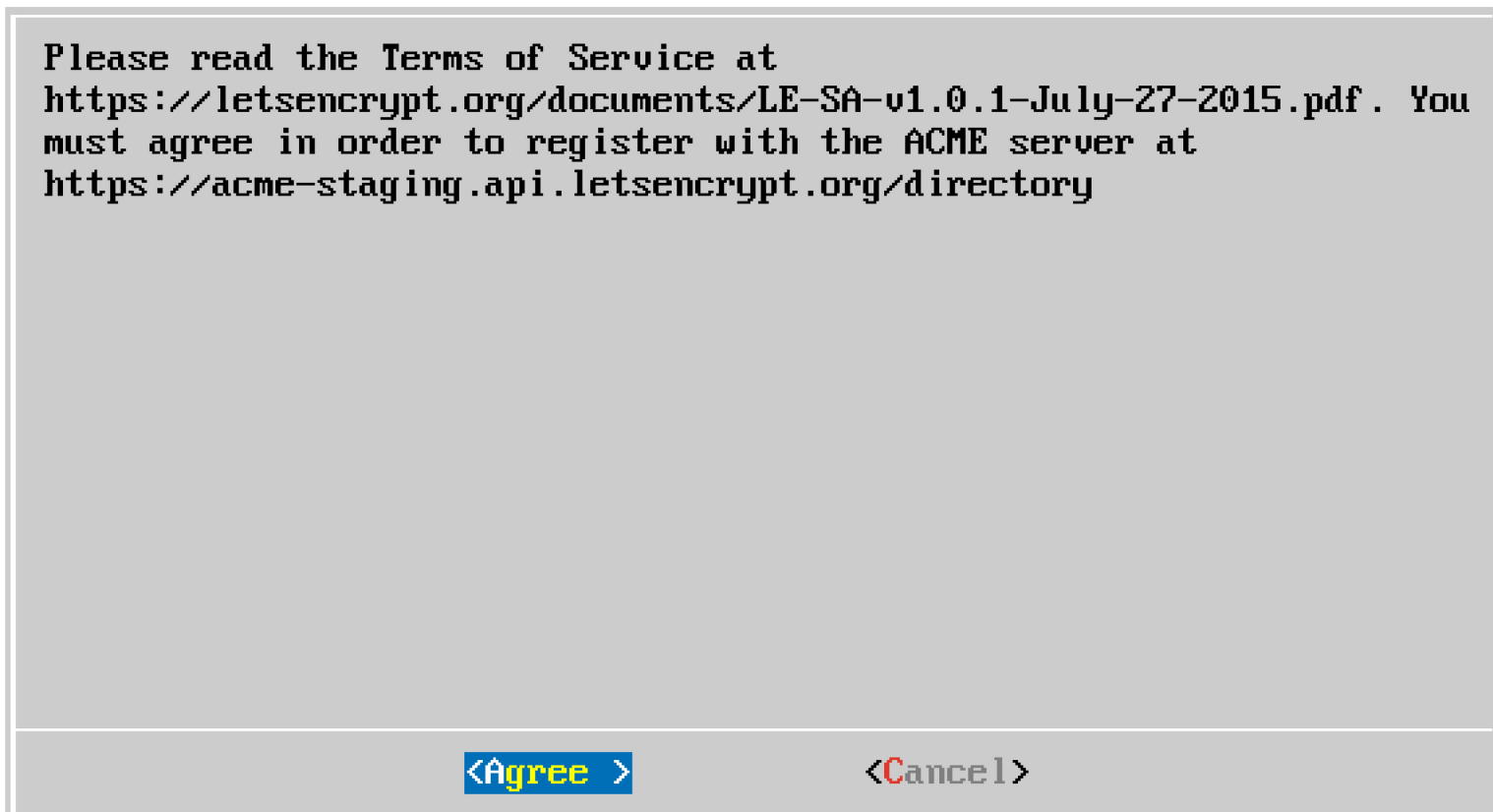
② 実行(証明書の取得とサーバーの自動設定)

Enter email address (used for urgent notices and lost key recovery)

< OK > <Cancel>

鍵復旧や緊急連絡の際の管理者メールアドレスを入力

② 実行(証明書の取得とサーバーの自動設定)



利用規約の同意で、<Agree>を選びリターン

② 実行(証明書の取得とサーバーの自動設定)

```
Congratulations! You have successfully enabled  
https://[redacted].com!  
  
You should test your configuration at:  
https://www.ssllabs.com/ssltest/analyze.html?d=[redacted].com
```

< OK >

証明書が無事発行され、ApacheのHTTPS設定も終わり<OK>を選択し終了

IMPORTANT NOTES:

```
- Congratulations! Your certificate and chain have been saved at  
/etc/letsencrypt/live/[redacted].com/fullchain.pem. Your cert will  
expire on 2016-01-23. To obtain a new version of the certificate in  
the future, simply run Let's Encrypt again.
```

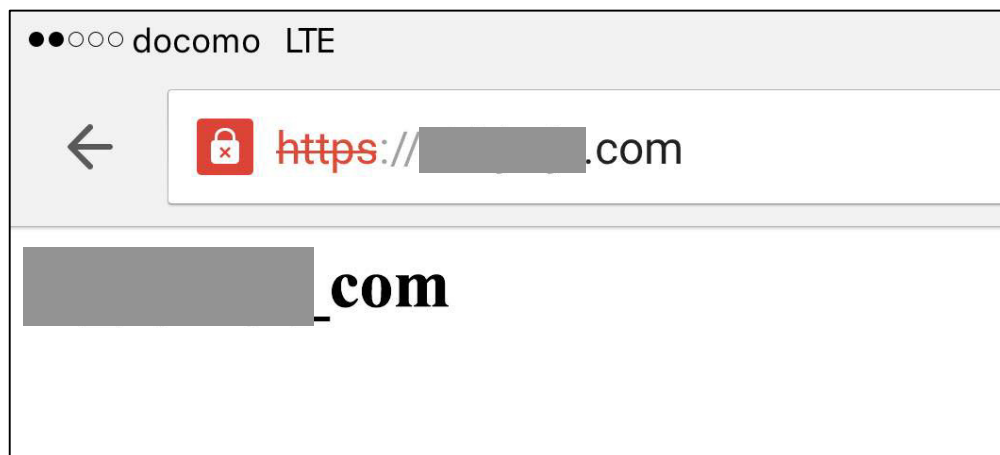
生成された鍵ペアや証明書、証明書チェーンの在り処
/etc/letsencrypt/live/ドメイン名/*.pem

初回はパッケージインストール等多少時間がかかるが
以降15~30秒で証明書発行、サーバ設定できる

早速アクセスしてみましょう



プレビューなので
テスト用CAから
発行された証明書で
警告が出ますが



証明書の設定がされ、
HTTPS接続できました

ベータプログラムや
一般公開後は
ちゃんとHTTPS接続可
能

生成されたファイルの置き場

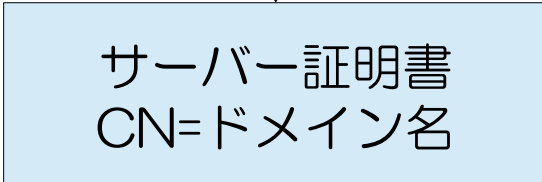
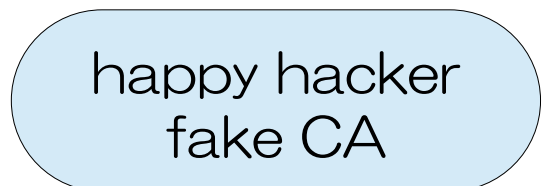
/etc/letsencrypt/live/ドメイン名

- **privkey.pem**
生成されたSSLサーバー証明書用の秘密鍵
- **cert.pem**
生成されたSSLサーバー証明書
- **chain.pem**
ルート証明書から全ての中間CA証明書まで
(プレビューではルートのみ)
- **fullchain.pem**
ルートからサーバー証明書まで全ての証明書チェーン

証明書チェーン

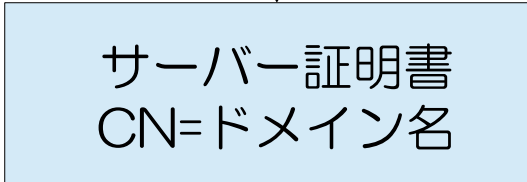
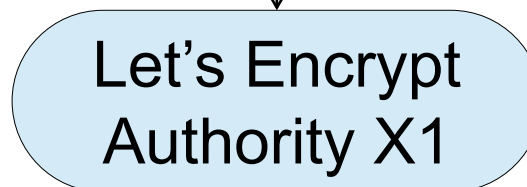
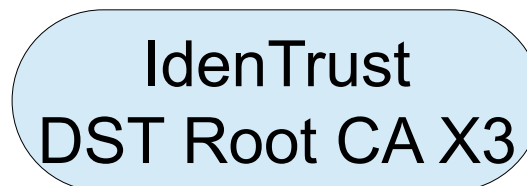
本番開始前
プレビュー

テスト用プライベートルート



ベータ/本番開始後
2015年11月末以降

パブリックルート



プレビューの証明書プロファイル(どんな形式?)

シリアル番号	18バイトランダム
署名アルゴリズム	SHA256withRSA
発行者	CN=happy hacker fake CA
有効期間	90日
主体者	CN=ドメイン名
鍵使用目的	DigSig, KeyEncipher
拡張鍵使用目的	TLS Server, TLS Client
主体者鍵ID	有り
発行者鍵ID	Keyidのみ
AIA拡張 (OCSP, CAIssuer)	http://ocsp.staging-x1.letsencrypt.org/ CAIssuer有
主体者別名	ドメイン名1, ...
証明書ポリシー	CPS= http://cps.letsencrypt.org/
CRL配布点	なし
基本制約	cA=FALSE

※ ECDSA証明書はツールが対応せず、発行できなそう。

ベータの証明書プロファイル(どんな形式?)

シリアル番号	18バイトランダム?
署名アルゴリズム	SHA256withRSA
発行者	Let's Encrypt Authority X1
有効期間	90日
主体者	CN=ドメイン名
鍵使用目的	DigSig, KeyEncipher
拡張鍵使用目的	TLS Server, TLS Client
主体者鍵ID	有り
発行者鍵ID	Keyidのみ
AIA拡張 (OCSP, CAIssuer)	http://ocsp.int-x1.letsencrypt.org/ CAIssuer有
主体者別名	ドメイン名1, ...
証明書ポリシー	CPS= http://cps.letsencrypt.org/
CRL配布点	なし
基本制約	cA=FALSE

※ ECDSA証明書はツールが対応せず、発行できなそう。

Let's Encrypt と Certificate Transparency (CT)

- CTとは、認証局が発行する全ての証明書の発行履歴を公開ログサーバーに登録するもので、何か不正があった場合でも、ログ調査可能にしている。
- Lets Encryptの全ての証明書発行の結果は、世界何箇所かにあるCTログサーバーに記録される。
- ただ、SCT対応でないため、Chromeで公開監査情報は見られない。



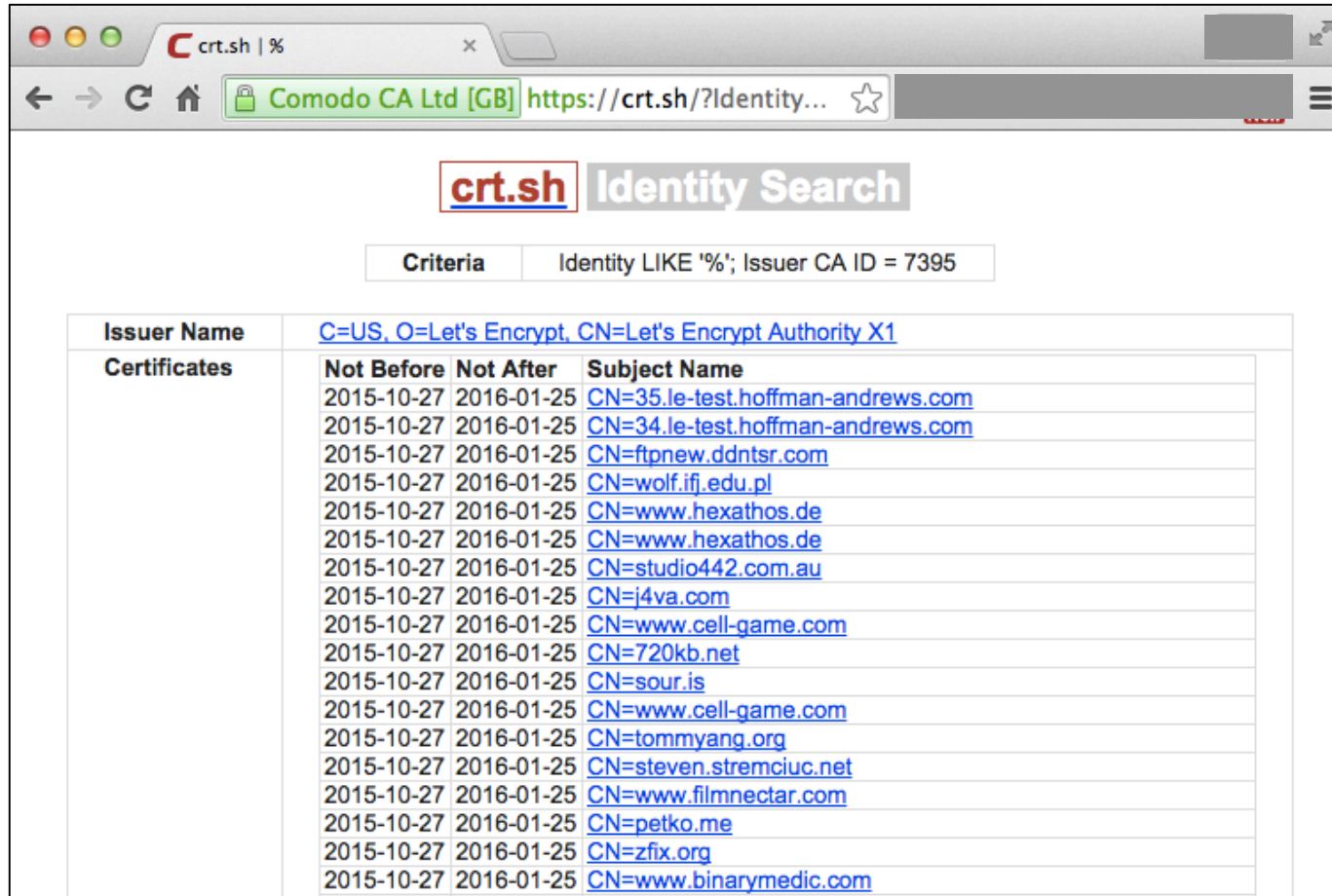
DigiCert, Inc. (所在地 US Utah Lehi) の識別情報は DigiCert SHA2 Extended Validation Server CA により確認済みで、公開監査が可能です。

[証明書情報](#)
[透明性に関する情報](#)

出典：
<http://www.symantec.com/ja/jp/page.jsp?id=ssl-certificate-transparency>

Let's Encrypt のCTのログ(1)

https://crt.sh/?Identity=%25&iCAID=7395



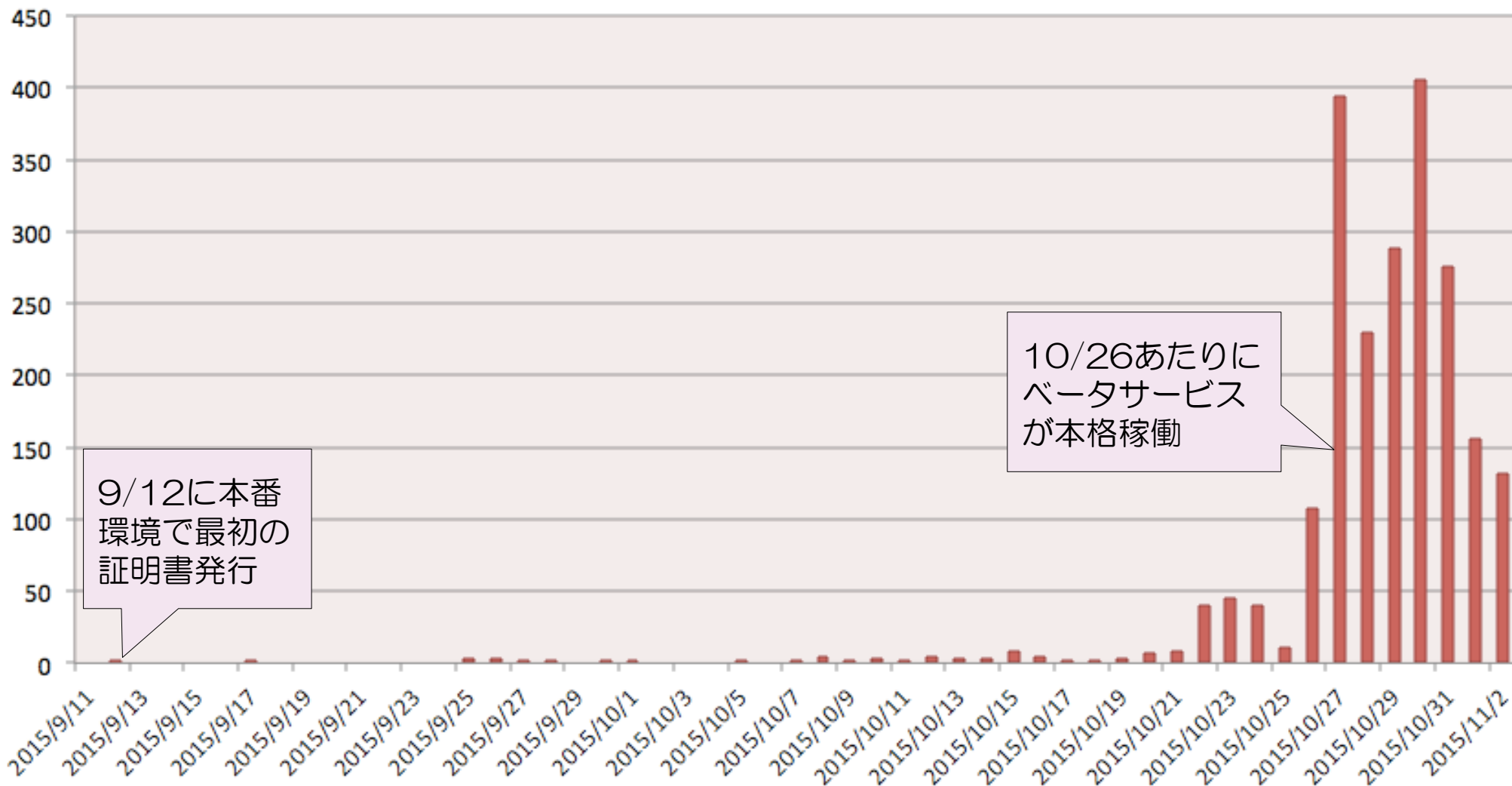
The screenshot shows a web browser window with the URL `https://crt.sh/?Identity=%25&iCAID=7395`. The page title is "crt.sh Identity Search". Below the title, there is a search criteria box containing "Identity LIKE '%'; Issuer CA ID = 7395". The main content is a table with the following data:

Issuer Name	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X1		
Certificates	Not Before	Not After	Subject Name
	2015-10-27	2016-01-25	CN=35.le-test.hoffman-andrews.com
	2015-10-27	2016-01-25	CN=34.le-test.hoffman-andrews.com
	2015-10-27	2016-01-25	CN=ftpnew.ddntrs.com
	2015-10-27	2016-01-25	CN=wolf.ifj.edu.pl
	2015-10-27	2016-01-25	CN=www.hexathos.de
	2015-10-27	2016-01-25	CN=www.hexathos.de
	2015-10-27	2016-01-25	CN=studio442.com.au
	2015-10-27	2016-01-25	CN=j4va.com
	2015-10-27	2016-01-25	CN=www.cell-game.com
	2015-10-27	2016-01-25	CN=720kb.net
	2015-10-27	2016-01-25	CN=sour.is
	2015-10-27	2016-01-25	CN=www.cell-game.com
	2015-10-27	2016-01-25	CN=tommyang.org
	2015-10-27	2016-01-25	CN=steven.stremciuc.net
	2015-10-27	2016-01-25	CN=www.filmnectar.com
	2015-10-27	2016-01-25	CN=petko.me
	2015-10-27	2016-01-25	CN=zfix.org
	2015-10-27	2016-01-25	CN=www.binarymedic.com

Let's Encryptで発行された証明書は全てCT公開監査ログに記載される

Let's Encrypt のCTのログ(2)

Let's Encryptの本番(ベータ含)証明書の日次発行枚数



Windows環境やTomcatなど サポート外環境ではどうする？

WindowsのIIS
Apache Tomcat
SSLアクセラレーター
Postfixなどのメールサーバー

などサポート外の環境ではどうしましょうか？



Let's Encrypt の証明書発行、HTTPS設定の流れ

ツールに含まれる暫定ウェブサーバーで証明書だけゲットします

準備

- Linuxで80/TCPが開いているサーバーの構築
- Gitでletsencryptツールのインストール

実行

- ツールの実行(証明書取得)
- サーバーの自動設定はしない

設定

- 必要があればPKCS#12(PFX)を作ります
- サーバーに証明書を設定します

サポート外SSLサーバー用の証明書取得(1)

“auth” コマンドでサーバーを設定せず証明書が取得できます

Apache、Nginxのサーバーは停止します
証明書を発行しようとしているホストであることを確認
コマンドラインでドメイン名は指定してしまいましょう。

```
% cd letsencrypt  
% sudo ./letsencrypt-auto auth -d w1.example.com
```

途中、暫定認証用のサーバーの選択では
“Standalone” を選んで下さい。

秘密鍵と証明書が生成され、ディレクトリに置かれます

```
/etc/letsencrypt/live/w1.example.com/*.pem
```

サポート外SSLサーバー用の証明書取得(2)

他のホストで動いているApache、Nginx、Lighttpd、Postfixなどは、そのままの鍵や証明書が使えます。

WindowsのIISやExchangeやApache Tomcatで証明書と鍵を使用する場合には、OpenSSLで、鍵と証明書チェーンを一つにしてパスワードを付けたPKCS#12 (or PFX)に変換して、インポートや設定を行います。

```
% openssl pkcs12 -export -in cert.pem -inkey privkey.pem  
-certfile fullchain.pem -out P12ファイル名  
-passout pass:パスコード
```

Windowsならファイルを開いてインポートして使用、TomcatならP12ファイルをkeystoreに設定し使用。

サーバー証明書の再発行と失効

コマンドライン一発でできる

失効

revokeオプションで、証明書を指定すれば、いつでも証明書を失効させることができる。

```
% sudo ./letsencrypt-auto revoke --cert-path cert.pem
```

再発行/更新

発行時と同じオプション(or 入力)をすると証明書を更新できる。有効期間3ヶ月なら、2ヶ月目程度でcronで再発行するのがオススメ。

```
% sudo ./letsencrypt-auto auth -d www.example.com
```

ワイルドカードとマルチドメイン(UC)証明書

コマンドライン一発でできる

ワイルドカード証明書

残念ながら、Lets Encryptではワイルドカード証明書(例 “*.example.com”)は発行できない。

マルチドメイン(UC)証明書

複数のドメインが同じサーバーにホスティングされている場合、(外部からWebアクセスして同じサーバーに辿着く場合)マルチドメイン(UC)証明書が発行できる。単に -d オプションで複数ドメインを記載すれば、subjectAltName拡張に全て記載される。

```
% sudo ./letsencrypt-auto auth -d www.example.com ¥  
-d t1.example.com -d example.com
```


Lets Encryptで自動設定されるApacheのHTTPS設定

```
/etc/letsencrypt/options-ssl-apache.conf
```

```
SSLProtocol all -SSLv2 -SSLv3  
SSLCipherSuite ECDHE-.. 後述  
SSLHonorCipherOrder on  
SSLCompression off
```

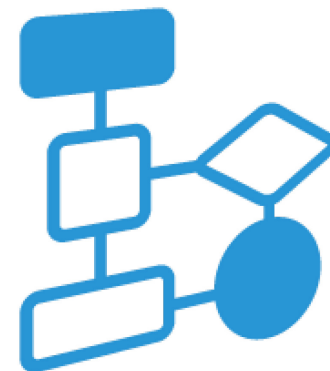
TLSv1.x のみ利用可
暗号スイートの選択や順序などかなりまとも
サーバーが示した暗号スイート優先順位に従う
CRIME攻撃等の対策としてデータ圧縮は無効化

設定の特徴

- 最近トレンドのHTTPS設定になっている。
- 最近話題の弱い暗号(RC4, EphemeralでないDH/ECDH, DES, MD5)は無効化
- 特にこの設定から大きくいじる必要はない。
- HSTS、Certificate Pinning、OCSP Staplingが必要なら追加設定

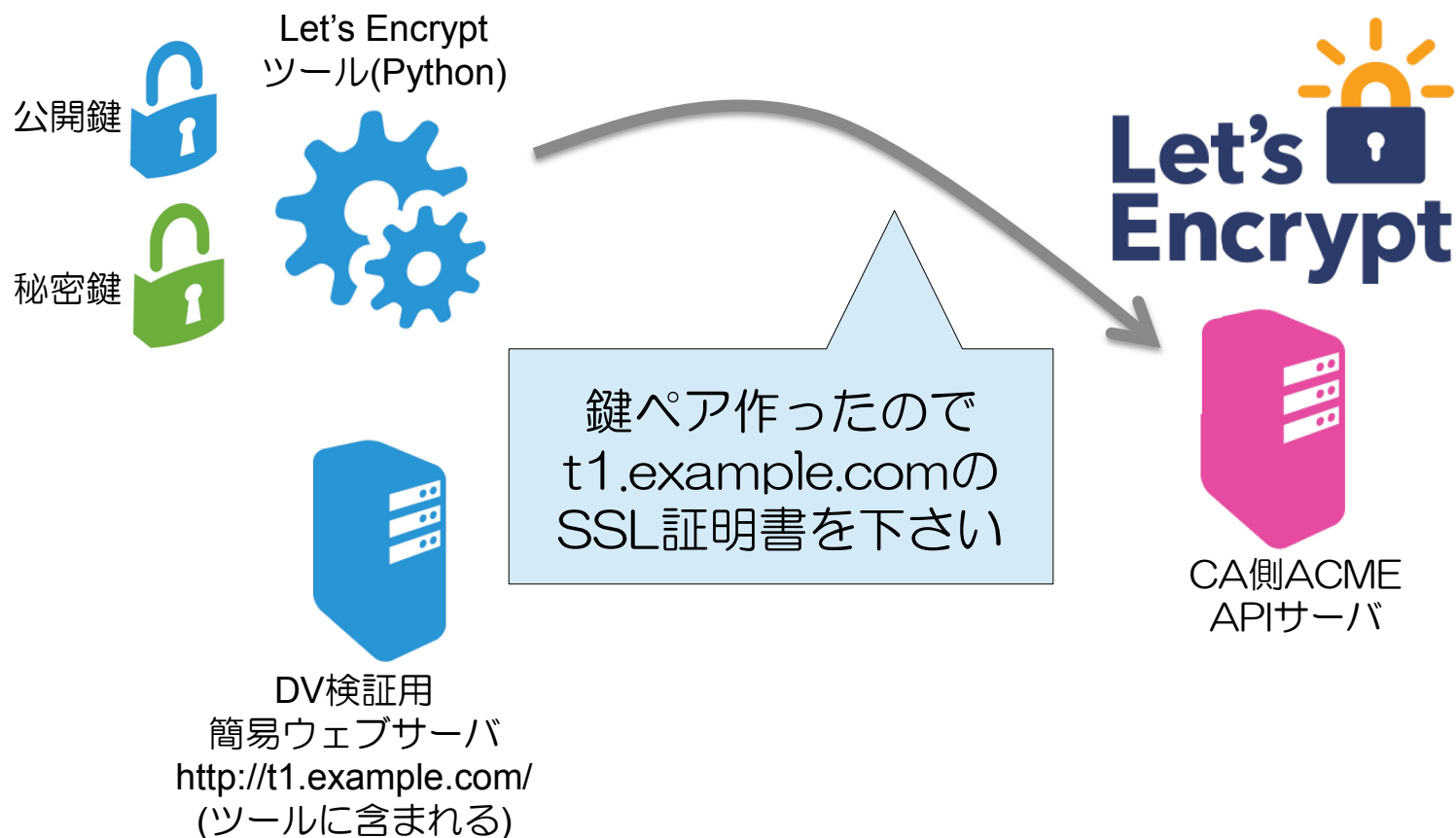
- 共通鍵の優先順位：AES128GCM, AES256GCM, AES, Camellia, 3DES
- 鍵交換の優先順位：ECDHE, DHE, RSA, SRP
- 認証の優先順位：RSA, ECDSA, DSS
- HMACの優先順位：SHA384, SHA256, SHA1

自動証明書発行のフロー (概要)



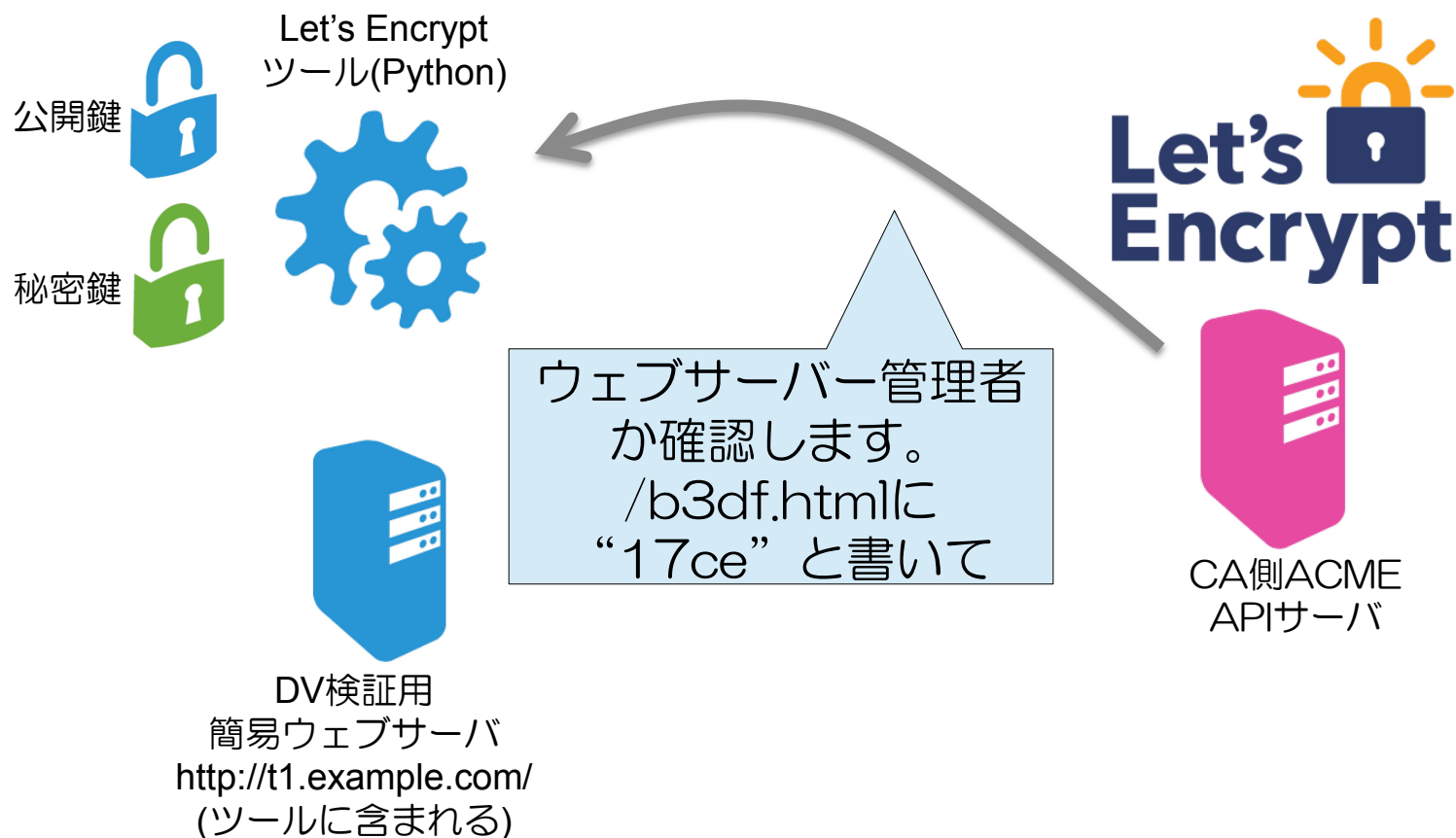
Let's Encrypt のドメイン検証(DV)

IETF I-DになっているACMEプロトコルで
ドメイン検証、証明書発行する。
メッセージはJSONベースでJWK、JWSなど使う



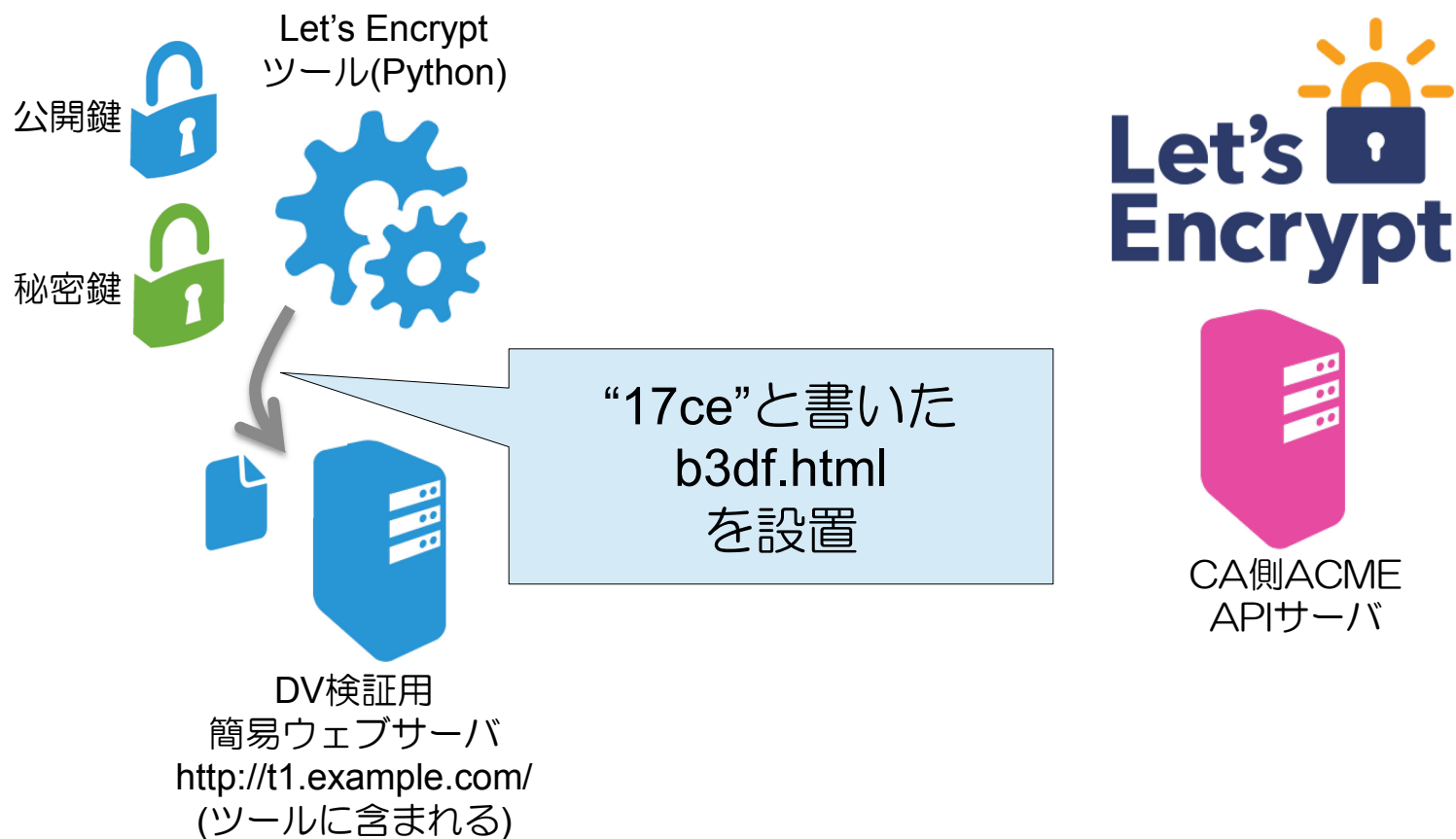
Let's Encrypt のドメイン検証(DV)

対象ドメインのウェブサーバーの
管理者かどうか確認する



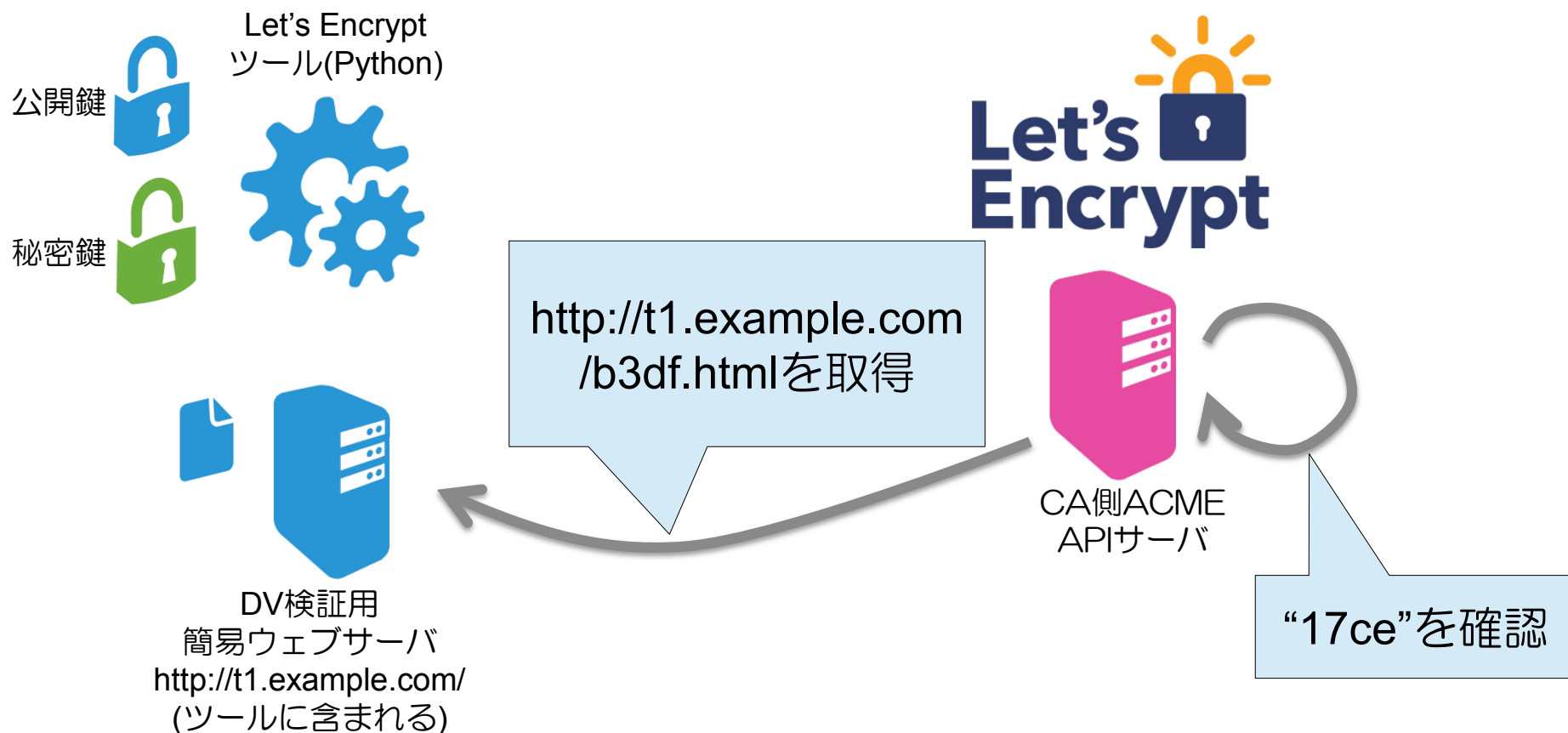
Let's Encrypt のドメイン検証(DV)

対象ドメインのウェブサーバーの
管理者かどうか確認する



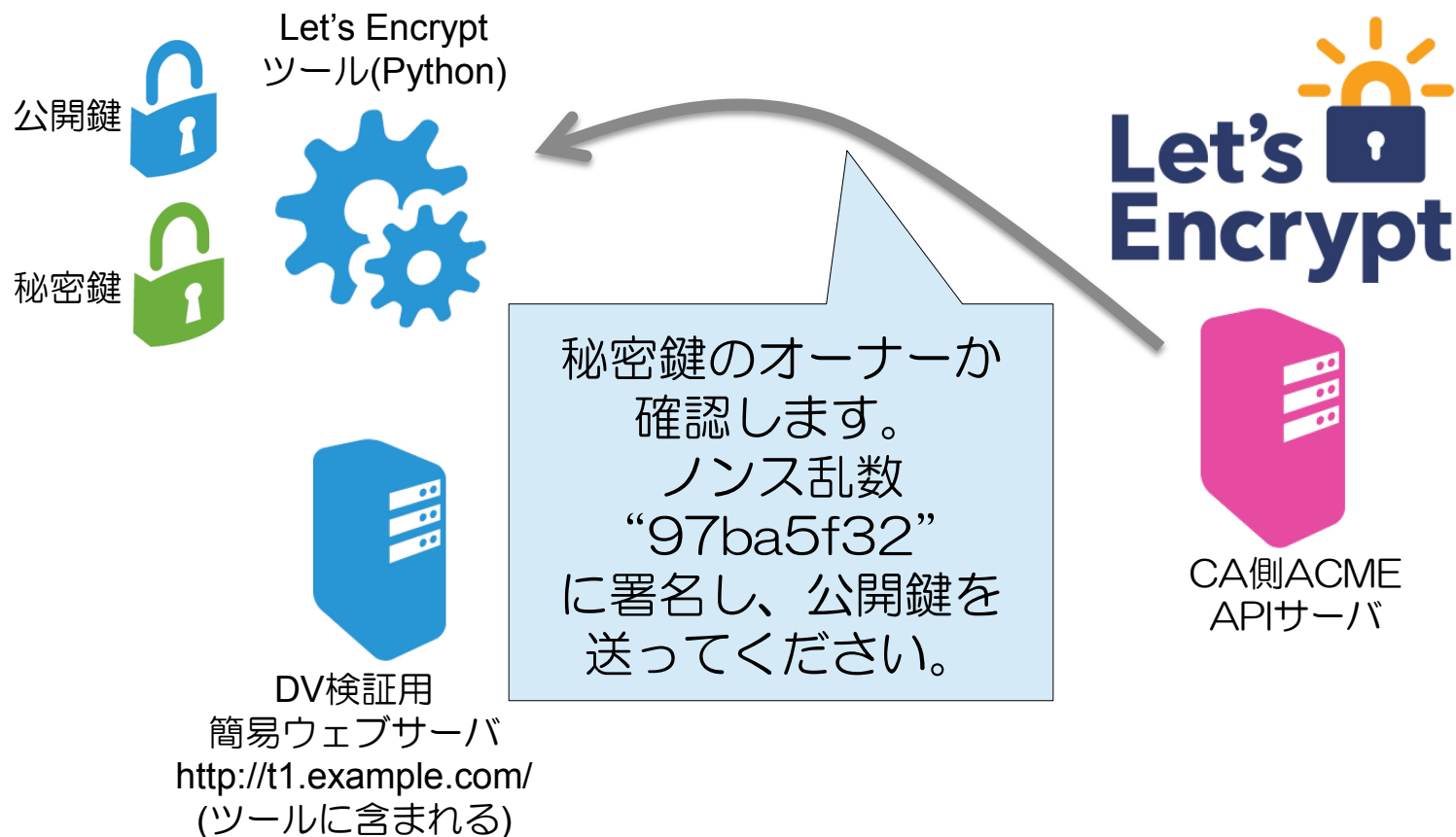
Let's Encrypt のドメイン検証(DV)

対象ドメインのウェブサーバーの
管理者かどうか確認する



Let's Encrypt のドメイン検証(DV)

次に、秘密鍵のオーナーであることを確認する



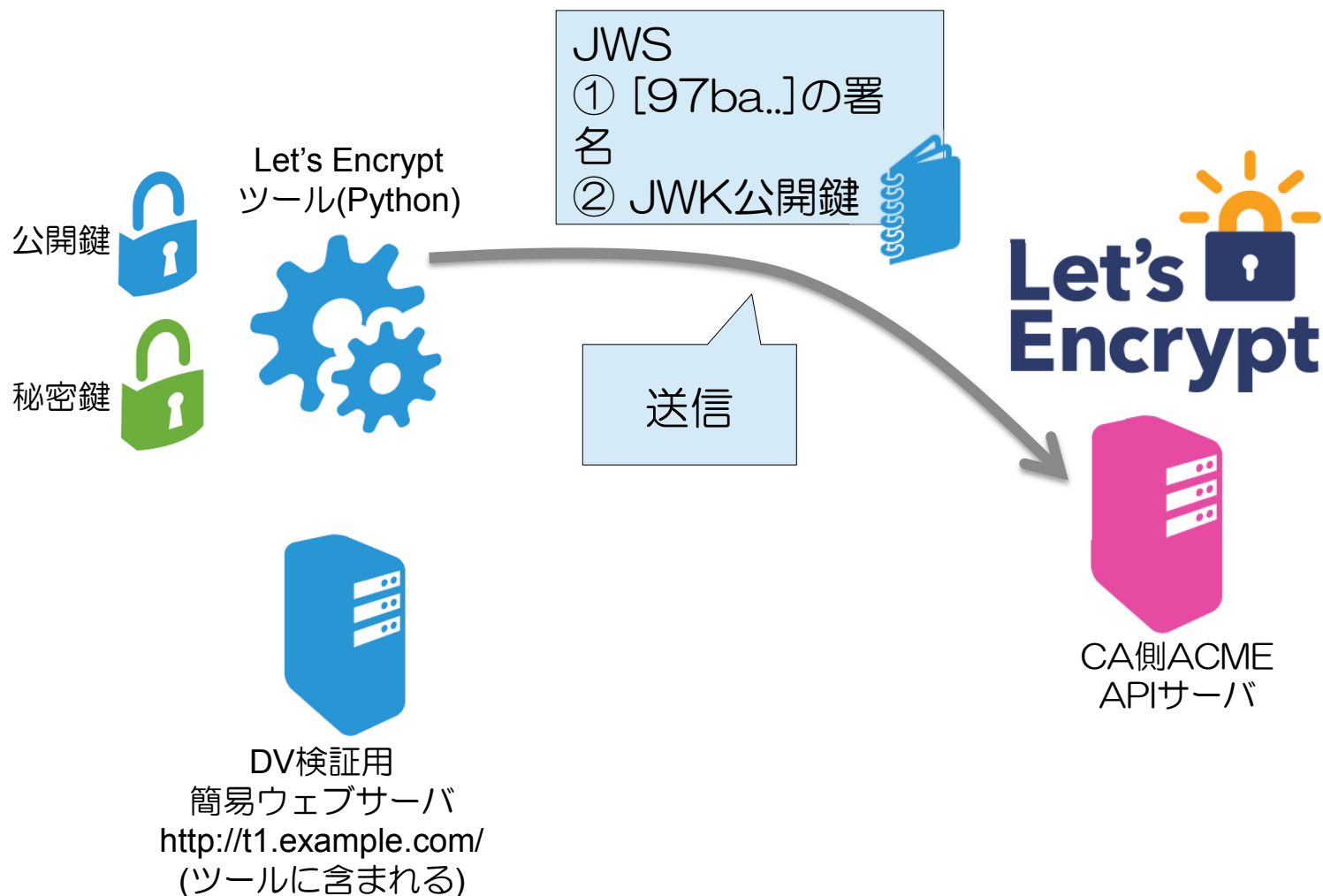
Let's Encrypt のドメイン検証(DV)

次に、秘密鍵のオーナーであることを確認する



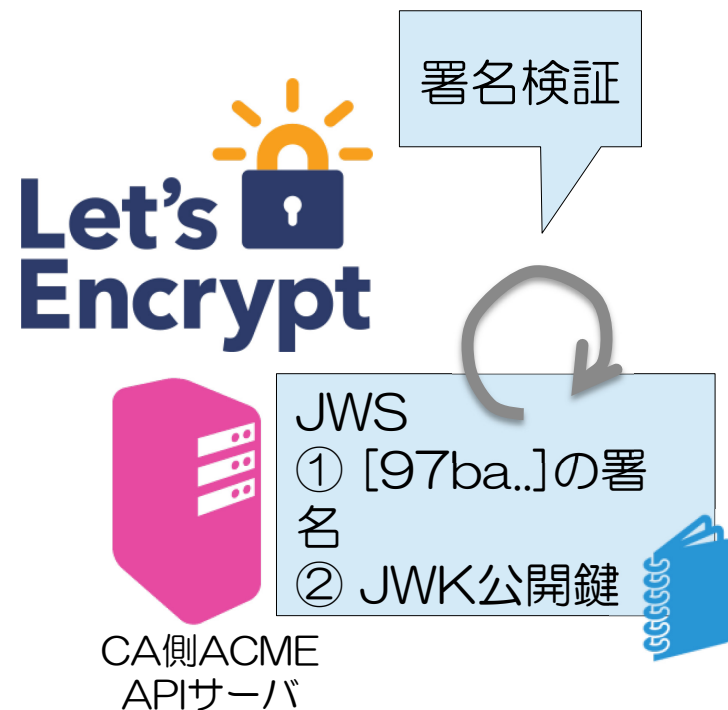
Let's Encrypt のドメイン検証(DV)

次に、秘密鍵のオーナーであることを確認する



Let's Encrypt のドメイン検証(DV)

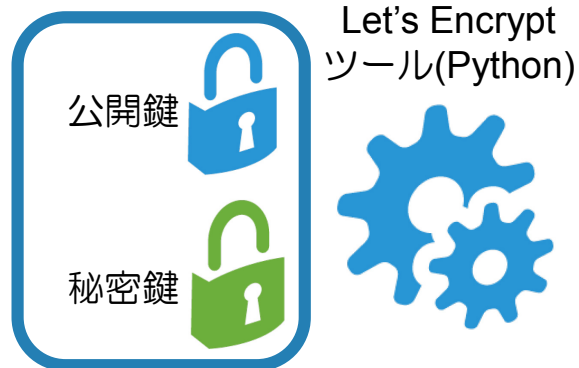
次に、秘密鍵のオーナーであることを確認する



Let's Encrypt のドメイン検証(DV)

ドメインの検証(DV)が完了

これをt1.example.com
ドメイン用に「認証された
鍵ペア」と呼ぶ



DV検証用
簡易ウェブサーバ
<http://t1.example.com/>
(ツールに含まれる)



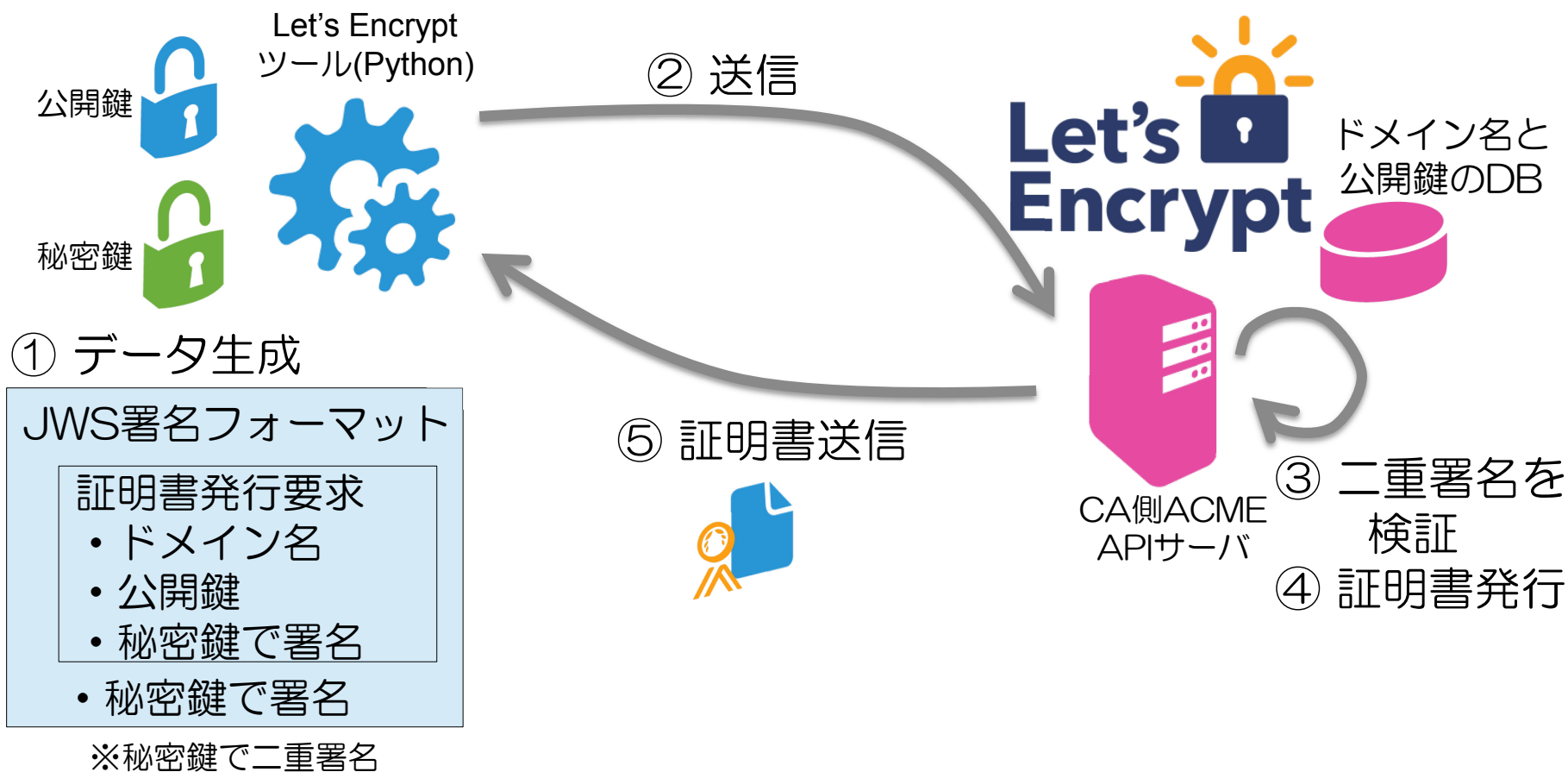
CA側ACME
APIサーバ



- ① ドメインのウェブサーバーの管理権限を持つ
 - ② ドメインに紐付いた秘密鍵の管理権限を持つ
- 以上を以って、ドメインの検証(DV)は完了

Let's Encrypt の証明書発行

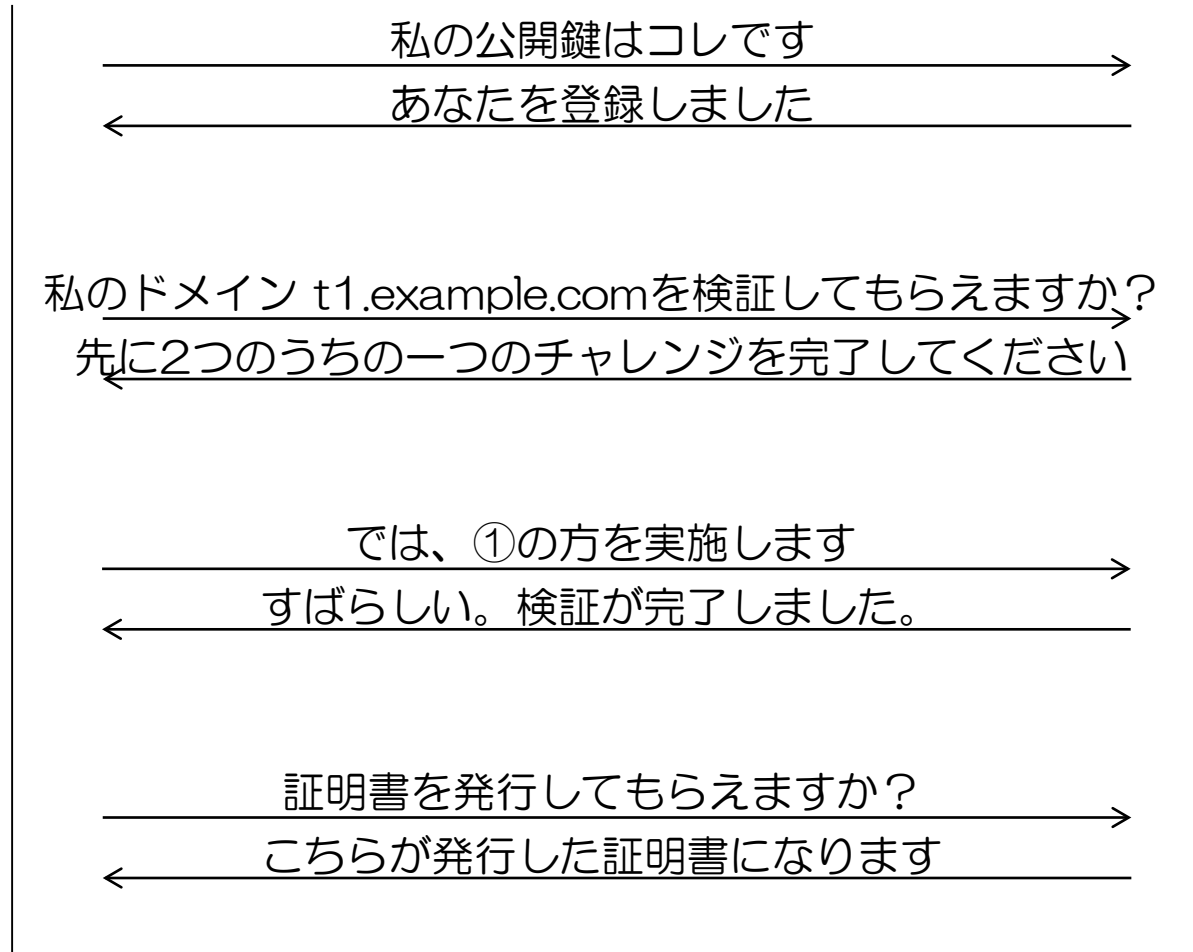
証明書発行要求(CSR/PKCS#10)を含む
JWS署名メッセージの生成と送信



ACMEプロトコルのフロー(概要)

Let's Encrypt
クライアント

boulder (CA)



まとめ

- SSL/TLSのおさらい
- 2015年のSSL/TLS関連に関連した話題
- Let's Encrypt！無料で簡単な証明書
- Let's Encrypt は開発用、個人用にはとても便利。特にサーバーの設定込み、バッチでもできるので便利。
- ただ、会社などで使う場合には、DVであることを制限を考えて利用してください。

