

ネットワークセキュリティ詳説  
**PKI/TLS プロトコル**

加藤 聰彦【著】

コロナ社

# まえがき

今日インターネットは、電気や水道などと並んで、社会生活に欠くことのできないインフラストラクチャとなっている。その利用は、単なる情報検索や通信にとどまらず、ネットショッピングやネットバンキングなどの経済活動にまで広がっている。一般の利用者からは、インターネットは、「つねに側にあって利用できるもの」となっている。

一方、技術的な側面では、インターネットの機能は大規模化、複雑化を続けている。インターネット関連の標準化を進めている IETF では、2022 年 2 月の時点で、9 000 を超える標準文書 (RFC) を定めている。また、無線 LAN の規格である IEEE 802.11 標準は、2020 年版が 4 000 ページを超えるサイズとなった。インターネットがインフラ化する裏で、その技術をフォローすることが困難となっている。

インターネットにおいて、技術更新が頻繁かつ注目されている分野の一つが、ネットワークセキュリティである。情報の流出や改ざんなどに対する多様な攻撃に対処するために、セキュリティ機能の改良、追加、大幅なバージョンアップなどが繰り返されている。本書では、ネットワークセキュリティ技術の中で、“https” で始まる URL を用いたセキュアなウェブサーバアクセスに使用される PKI (public key infrastructure : 公開鍵基盤) と TLS (transport layer security : トランスポートレイヤセキュリティ) に関して解説する。

本書を作成するにあたっては、できるだけ詳細な技術内容を取り込み、かつわかりやすく説明することを心掛けた。具体的には、PKI と TLS に関連する 70 以上の RFC などの文書を対象とし、踏み込んだ説明を行った。その中には、2018 年に規格化された TLS バージョン 1.3 や、2021 年に規格化された QUIC などの最新情報も含まれる。PKI/TLS の最新の技術動向を広くカバーできていると自負している。

本書の構成はつぎのような三つに分けられる。

- 1～3 章で、PKI/TLS で使用される情報セキュリティ技術の基礎を解説する。暗号化や認証の基本原則から、認証付き暗号 (AEAD)、RSASSA-PSS、楕円曲線暗号、エドワーズ曲線といった比較的新しい技術までを説明する。
- 4 章と 5 章が PKI に関する解説である。4 章で、認証局や証明書などに関して詳細に説明する。5 章では、証明書を検索・管理するための通信プロトコルを広く解説する。
- 6～8 章が TLS に関する解説である。現在普及している TLS バージョン 1.2、新たに導入されたバージョン 1.3、Google 社が主導的に導入している QUIC のセキュリティなどについて詳しく説明する。

本書では、複数の読者層を想定している。まず、PKI/TLS について学習する読者である。全体を読み込むことにより、専門的な知識を得ることができると考えている。つぎは、セキュリティ初心者である読者である。各章・各節の説明は、概要から詳細に移行するよう努めている。概要の部分に着目することにより、大まかな内容を理解できると思う。最後に、ネットワーク技術者による辞典的な使い方である。PKI や TLS に関して特定のトピックを調べる場合、例えば、OCSP や TLS 1.3 について知りたい、証明書や TLS の特定のエクステンションの意味がわからない、などの場合は、それに関する部分を読むことで理解が進むと思う。このため、各章は独立して読めるように、また目次や索引が多くのキーワードを指定するように努めている。

本書がインターネットにおけるセキュリティの技術動向を理解する一助になればと願っている。

2022 年 7 月

加藤 聡彦

# 目 次

## 1. 情報セキュリティ技術の基本

1.1	ウェブサーバアクセスのためのセキュリティ技術	1
1.2	慣用系暗号方式	3
1.2.1	概 要	3
1.2.2	ストリーム暗号とブロック暗号	4
1.3	公開鍵暗号方式	6
1.4	認 証	7
1.5	鍵 管 理	9
1.5.1	慣用系暗号方式における鍵管理	9
1.5.2	公開鍵暗号方式における鍵管理	9

## 2. 慣用系暗号のセキュリティ技術

2.1	ストリーム暗号	11
2.1.1	RC4	11
2.1.2	ChaCha20	12
2.2	ブロック暗号	15
2.2.1	DES	15
2.2.2	トリプルDES	18
2.2.3	AES	18
2.2.4	利用モード	21
2.2.5	パディング	22
2.3	ハッシュ関数	23
2.3.1	MD5	24
2.3.2	SHA-1とSHA-2	25
2.4	メッセージ認証コード	27
2.4.1	HMAC	27
2.4.2	ブロック暗号によるMAC	28
2.4.3	Poly1305	28
2.5	認証付き暗号 AEAD	29

2.5.1	CCM	30
2.5.2	GCM	31
2.5.3	AEAD_CHACHA20_POLY1305	32
2.6	鍵 共 有	33
2.6.1	RSA による鍵共有	33
2.6.2	Diffie-Hellman 鍵共有	34
2.6.3	Diffie-Hellman 鍵共有の現実的な課題	35
2.6.4	SRP プロトコル	37

### 3. 公開鍵暗号のセキュリティ技術

3.1	RSA 暗 号	40
3.1.1	アルゴリズム	40
3.1.2	RSA 暗号による暗号化とデジタル署名	41
3.1.3	実際の暗号化処理	42
3.1.4	実際の署名処理	45
3.2	ElGamal 暗 号	47
3.2.1	アルゴリズム	47
3.2.2	DSA	48
3.3	楕円曲線暗号	49
3.3.1	概 要	49
3.3.2	鍵共有とデジタル署名	51
3.3.3	エドワーズ曲線	52
3.3.4	EdDSA	53

### 4. 公開鍵基盤：PKI

4.1	PKI の モ デ ル	55
4.1.1	概 要	55
4.1.2	証明書の基本	56
4.1.3	証明書の失効	60
4.2	証 明 書	61
4.2.1	証明書の構造	61
4.2.2	RFC 5280 の定めるエクステンション	64
4.2.3	証明書の透明性	71
4.3	署名アルゴリズムおよび公開鍵	73
4.3.1	ハッシュ関数	73

4.3.2	署名アルゴリズム	73
4.3.3	公開鍵の情報	74
4.4	証明書の失効リスト	77
4.4.1	概 要	77
4.4.2	CRL の 構 造	77
4.4.3	CRL エクステンション	79
4.4.4	CRL エントリエクステンション	81

## 5. PKI のための通信プロトコル

5.1	PKI 操作プロトコルの概要	83
5.2	LDAP による PKI 操作プロトコル	84
5.2.1	概 要	84
5.2.2	LDAP の情報モデル	84
5.2.3	LDAP による PKI 情報の管理	86
5.2.4	LDAP のメッセージ	88
5.3	FTP/HTTP による PKI 操作プロトコル	91
5.4	PKI 管理プロトコルの概要	92
5.4.1	管理プロトコルの種類	92
5.4.2	管理プロトコルの機能	92
5.4.3	証明書の初期登録	93
5.4.4	証明書・鍵ペアの更新	94
5.4.5	鍵ペアの回復	94
5.4.6	証明書の失効の要求	95
5.4.7	クロス証明書の初期登録・更新	95
5.4.8	ルート証明書の更新	95
5.4.9	証明書や CRL の公開	95
5.5	PKI 管理プロトコル CMP	96
5.5.1	概 要	96
5.5.2	メ ッ セ ー ジ	97
5.5.3	通 信 手 順	100
5.6	PKI 管理プロトコル CMC	104
5.6.1	概 要	104
5.6.2	単純なトランザクション：simple PKI request / response	105
5.6.3	フルスペックのトランザクション：full PKI request / response	106
5.7	証明書登録プロトコル SCEP / EST	111
5.7.1	SCEP	111
5.7.2	EST	114

5.8	OCSP	116
5.8.1	概 要	116
5.8.2	プロトコルの詳細	118
5.9	SCVP	120
5.9.1	概 要	120
5.9.2	証明書検証リクエスト	121
5.9.3	証明書検証レスポンス	123

## 6. トランスポートレイヤセキュリティ：TLS

6.1	TLS 1.2 までの経緯	125
6.2	TLS の モデル	126
6.2.1	基本的アイデア	126
6.2.2	レイヤ構成	126
6.2.3	要 点	127
6.2.4	PRF 関数と鍵の生成	129
6.3	TLS レコードプロトコル	130
6.3.1	概 要	130
6.3.2	MAC 計算および暗号化のためのデータ構造	132
6.4	TLS ハンドシェイクプロトコル	133
6.4.1	概 要	133
6.4.2	暗号スイート	135
6.4.3	詳 細 手 順	141
6.4.4	TLS-SRP	157
6.5	DTLS	159
6.5.1	概 要	159
6.5.2	設 計 方 針	159
6.5.3	レコードプロトコル	160
6.5.4	ハンドシェイクプロトコル	161

## 7. TLS バージョン 1.3

7.1	TLS 1.2 との違い	164
7.2	通信シーケンス	165
7.2.1	フルハンドシェイク	165
7.2.2	セッションの再開	168
7.2.3	0-RTT モード	170

7.3	HKDF 関数と鍵の生成	170
7.3.1	HKDF 関数	170
7.3.2	鍵の生成	171
7.4	ハンドシェイクプロトコルの詳細	173
7.4.1	概要	173
7.4.2	鍵交換メッセージ	174
7.4.3	エクステンション	176
7.4.4	サーバパラメータメッセージ	182
7.4.5	認証メッセージ	183
7.4.6	EndOfEarlyData メッセージ	184
7.4.7	ハンドシェイク後メッセージ	185

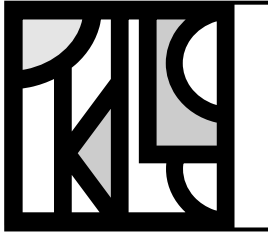
## 8. QUIC セキュリティ

8.1	概要	187
8.2	プロトコル手順	190
8.2.1	パケットとフレーム	190
8.2.2	通信シーケンス	193
8.3	セキュリティ機能	195
8.3.1	鍵の生成とパケットの保護	195
8.3.2	TLS 1.3 との差異	196
8.3.3	通信の解析	197

引用・参考文献 201

索引 205





# 情報セキュリティ 技術の基本

PKI/TLS プロトコルの解説の第一歩として、本章では情報セキュリティ技術の基本を紹介する。具体的には、慣用系暗号方式、公開鍵暗号方式、認証、鍵管理についての概要を説明する。



## 1.1 ウェブサーバアクセスのためのセキュリティ技術



今日、インターネットを介した情報通信は、社会のインフラストラクチャの一つとなっており、ウェブサーバアクセスによるオンラインショッピングやオンラインバンキングなども日常的に行われている。しかしこれらの通信では、クレジットカードの番号や銀行口座の ID / パスワードなどの個人情報が盗まれ悪用されないように、細心の注意が必要となる。このようなウェブサーバアクセスを安全に実現するために、広く用いられている通信プロトコルが TLS (トランスポートレイヤセキュリティ) であり、ウェブサーバの正当性保証をサポートする仕組みが PKI (公開鍵基盤) である。現在では、多くのウェブサーバが PKI / TLS を使用している。

TLS 通信の例として、ブラウザ (Microsoft Edge) を用いて電気通信大学のウェブサーバにアクセスした結果を図 1.1 に示す。アクセス先の URL (uniform resource locator) は “https://www.uec.ac.jp” (図中①) であり、プロトコルの種類を示す先頭の “https” が、TLS を用いて HTTP (hypertext transfer protocol) の通信を行うことを示す。また URL の表示の左側に鍵のマークがあるが (図中②)、これはセキュリティ技術により安全な通信を行っていることを示す。実際、この部分をクリックすると、「接続がセキュリティで保護されています」と表示さ



図 1.1 電気通信大学のウェブサーバへのアクセス結果

れる。

ウェブサーバアクセスの安全性を保障するためには、いくつかの要求条件がある。例えば、ウェブサーバにアクセスする利用者（クライアント）とサーバの間で転送されるデータは、悪意のある攻撃者が盗聴したとしても解読できないように暗号化されていなければならない。また、データが途中で改ざんされた場合は受信側で検知可能でなければならない。加えて、そもそも偽物でない正当なサーバと通信していることが保証されなければならない。これらを満足するために、TLS ではこれまでに情報セキュリティ技術として検討された多くのものを利用している。

そこで本書では、1～3章で、PKI/TLS で使用される情報セキュリティ技術について解説する。この部分では情報セキュリティの基本から、AEAD や楕円曲線暗号といった比較的最近に使用されるようになった技術まで、一通り詳解する。なお、それらの解説では数学的な詳細については触れていない。本章ではまず、基本的な概念について解説する。

そもそも情報セキュリティとは、情報の機密性、完全性、可用性を維持することと定義されている。**機密性**（confidentiality）とは、許可されていない個人、エンティティまたはプロセスに対して、情報を使用不可または非公開にする特性である。**完全性**（integrity）とは、情報の正確さおよび完全さを保護する特性である。そして**可用性**（availability）とは、許可されたエンティティが要求したときに、情報のアクセスおよび使用が可能であるという特性である。本書では、PKI/TLS について詳細に解説するために、機密性と完全性に着目する。

**暗号化**（encryption）とは、情報を第三者に盗み見られないように、すなわち機密性を保証するように、決まった規則に従って変換することである。図 1.2 に示すように、暗号化を行う前の情報を<sup>ひら</sup>平文（plaintext）と呼び、暗号化された後の情報を**暗号文**（ciphertext）と呼ぶ。また、暗号文を平文に直す作業を**復号**（decryption）と呼ぶ。

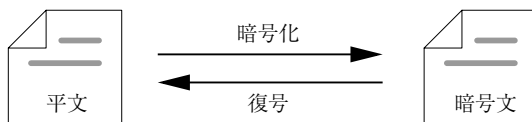


図 1.2 暗号化と復号

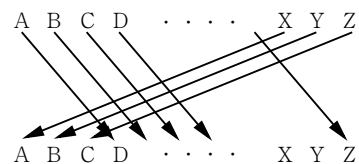


図 1.3 シーザー暗号

図 1.3 にジュリアス・シーザーが使用したといわれる**シーザー暗号**（Caesar cipher）の例を示す。これは、アルファベットを一定数だけずらすことにより暗号化し、逆に戻すことにより復号するというものである。図の例では、3文字ずつずらしている。これにより、「TOSHIHIKO KATO」という平文は「WRVKLKLNR NDWR」という暗号文に変換される（ただし空白の処理は省いている）。情報の送受信者が「3文字ずらす」ということを知っている場合に、正確かつ安全な情報の伝達が可能となる。

これは非常に単純な方式であるが、現在の暗号方式につながる重要な性質を持っている。そ

これは、暗号化および復号のアルゴリズムは既知であり、「何文字ずらすか」という情報のみが秘密だということである。このような、暗号化および復号に使用する秘密の情報を鍵 (key) と呼ぶ。アルゴリズムを公開し、鍵のみを秘密にすることにより、暗号のソフトウェアを誰もが開発できる、暗号の設計者が悪意のある仕掛けを施していないことを確認できる、などの利点が生じる。一方、鍵の情報が知られただけで暗号が破られる、という欠点もある。実際、シーザー暗号は鍵の種類が 26 個しかないため、容易に解読可能である。

シーザー暗号は機密性の保証のみを目的とするが、前述のように、現在の情報セキュリティにおいては、完全性の保証も重要な目的の一つとなっている。完全性の保証とは、具体的にはつぎの二つの攻撃への対応である (図 1.4 参照)。第一の攻撃は、正しい送信者 A が受信者に対して情報を送信した際に、攻撃者がその内容を改ざんするというものである。第二は、攻撃者 A' が、送信者 A と偽って受信者に情報を送信するというものである。これらに対して、完全性の保証により、情報が正しい作成者によって作成され、かつその情報が悪意のある第三者によって改ざんされていないことが保証される。この完全性の保証を実現するための処理を認証 (certification) と呼ぶ。認証では、対象となる情報を特徴づける付加情報を暗号技術により生成し、元の情報と付加情報の対により、情報の発信者が想定した者であること、元の情報 (および付加情報) が改ざんされていないことの二つを保証する。

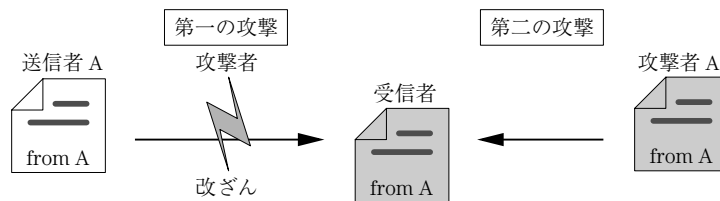


図 1.4 完全性の保証が想定する攻撃

また、前述のシーザー暗号では、暗号化と復号に同じ鍵「3文字ずらす」を使用した。このような暗号方式を慣用系暗号方式と呼ぶ。これに対して、暗号化と復号に異なる鍵を使用し、そのうち一方のみを公開するという公開鍵暗号方式も存在する。

以下では、慣用系暗号方式と公開鍵暗号方式、認証、鍵の管理について順に概説する。



## 1.2 慣用系暗号方式



### 1.2.1 概要

慣用系暗号方式 (conventional cryptosystem) とは、図 1.5 に示すように、平文を暗号化する側と、暗号文を復号する側が、同一の鍵を用いる方式である。この方式は、共通鍵暗号方式 (common key cryptosystem)、秘密鍵暗号方式 (secret key cryptosystem)、対称鍵暗号方式 (symmetric key cryptosystem) などとも呼ばれる。使用される鍵は、暗号化する側と復号する



図 1.5 慣用系暗号方式

側のみが保持しており，第三者には知らせてはならない。このため，**秘密鍵** (secret key) と呼ばれる。

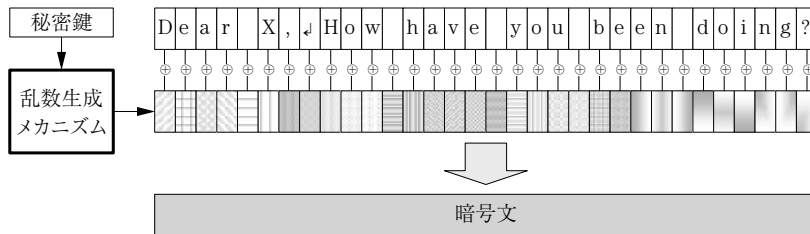
この方式では， $E(k, M)$  を暗号化の関数， $D(k, M)$  を復号の関数とすると，暗号化と復号の関係は次式で与えられる。ただし， $M$  は暗号化の対象となる情報（平文または暗号文）， $k$  は秘密鍵である。

$$M = D(k, E(k, M))$$

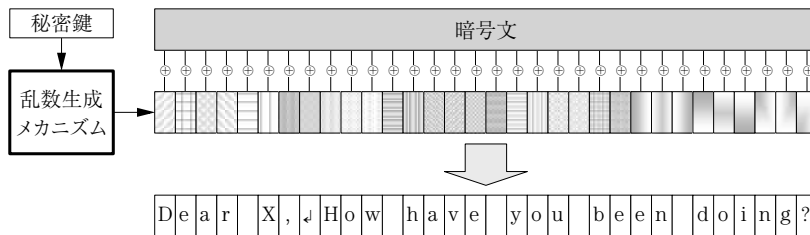
慣用系暗号方式は，ハードウェアによる実現を想定した方式であり，高速な暗号化・復号の処理が可能である。そこで，転送するパケットの暗号化など，比較的長いデータの処理を高速で行う場合などに用いられる。

### 1.2.2 ストリーム暗号とブロック暗号

転送するパケットの暗号化において，対象となる平文は可変長である。このような状況に対応するために，慣用系暗号方式ではつぎの二つのアプローチを採用している。第一は，平文と同じ長さのランダムなバイト列（乱数列）を生成し，その乱数列を用いて暗号化するストリー



(a) 暗号化



(b) 復号

図 1.6 ストリーム暗号の概要

ム暗号 (stream cipher) である。図 1.6 にその概要を示す。図 (a) の暗号化においては、秘密鍵に基づいて、平文と同じ長さの乱数列を生成し、平文のバイトと対応する乱数列のバイトとの排他的論理和を計算して暗号文とする。図 (b) の復号においては、同じ乱数生成メカニズムと秘密鍵を用いて、暗号化と同一の乱数列を生成する。この乱数列と暗号文との各バイトの排他的論理和を計算することにより、平文を復元する。

第二のアプローチは、平文を固定長のブロックに分割し、ブロックごとに暗号化および復号を行うブロック暗号 (block cipher) である。その概要を図 1.7 に示す。平文は固定長 (図の例では 16 バイト) に分割され、ブロックごとに秘密鍵を用いて暗号化・復号される。なお、この例では平文がブロック長の整数倍であるが、平文の最後の部分はブロック長よりも短くなるのが一般的である。そのような場合は、暗号化時に最後のブロックに追加の情報を埋め込む必要がある。



(a) 暗号化



(b) 復号

図 1.7 ブロック暗号の概要

また、平文が複数のブロックに分割される場合は、連続したブロックをどのように関連付けるかを考慮する必要がある。このメカニズムをブロック暗号の利用モードと呼ぶ。図では各ブロックを独立に暗号化・復号しているが、より安全性を高めるために、多くの方法が提案されている。

ストリーム暗号の代表例としては、RC4が挙げられる。またブロック暗号としては、かつては米国の国家暗号規格であったDESが代表例だった。しかし、その脆弱性が指摘され、現在はAESが広く用いられている。

# 索引

## 【あ】

アラートプロトコル	127
暗号化	2
暗号スイート	127
暗号セッション	127
暗号文	2

## 【い】

位数	50
委任されたパス検証	121
委任されたパス発見	121

## 【え】

エクステンション	56,64
エドワーズ曲線	52
エドワーズ曲線デジタル署名 アルゴリズム	53,165
エニーポリシー	67
エニーポリシー抑制	70
エポック	160
エンドエンティティ	55
エンドエンティティ証明書	57

## 【お】

オブジェクトエントリ	84
オブジェクトクラス	84
オブジェクト識別子	62

## 【か】

鍵	3
鍵共有	33
鍵交換フェーズ	166
鍵使用方法	66
鍵ペア	92
拡張鍵使用方法	69
拡張マスタシークレット	144
可変長整数表現	191
可用性	2
間接 CRL	77
完全 CRL	60,77
完全性	2
慣用系暗号方式	3
管理トランザクション	56,83

## 【き】

基準 CRL	77
--------	----

擬似乱数関数	128
基本制約	68
基本認証方式	100
機密性	2
共通鍵暗号方式	3

## 【く】

クライアント乱数	128
クロス証明書	57

## 【け】

権威者鍵識別子	64,79
権威者情報アクセス	70,81
検証された証明パス	122
検証データ	134
検証ポリシー	121
検証ポリシーリクエスト/ レスポンス	121

## 【こ】

公開鍵	6
公開鍵暗号方式	6
公開鍵情報	63
構造型オブジェクトクラス	85
コネクション ID	191

## 【さ】

最新 CRL	70,81
再ネゴシエーションハンド シェーク	145
サーバパラメータフェーズ	166
サーバ乱数	128
サブジェクト	55,63
サブジェクト鍵識別子	65
サブジェクト情報アクセス	71
サブジェクト代替名	68
サブジェクトディレクトリ属性	68
差分 CRL	60,77
差分 CRL 表示	80
差分 CRL 分配ポイント	70,81

## 【し】

識別名	62
シークレット	126
シークケンス番号	160
自己署名証明書	57

シーザー暗号	2
事前鍵共有	9
事前共有鍵	9,135
事前証明書	72
失効	55
失効した証明書	78
証明書	10,55
——の透明性	71
証明書管理プロトコル	92
証明書検証リクエスト/ レスポンス	121
証明書失効リスト	55,60,77
証明書チェイン	56
証明書テンプレート	96
証明書なし認証	115
証明書発行者	82
証明書ベース認証	115
証明書ポリシー	67
証明パス	57
証明パス検証	121
証明パス構築	121
省略型ハンドシェーク	134
初期化ベクタ	22,129
初期データ	170
初期認証鍵	96
ショートパケットヘッダ	190
署名アルゴリズム	61,62,77
署名値	64
シリアル番号	61
真正性	7
シンプル証明書登録プロトコル	92,111
信頼された SCVP サーバ	121

## 【す】

ストリーム	189
ストリーム暗号	4

## 【せ】

生成元	34
セキュア遠隔パスワード	37
セキュアトランスポート上の 登録	92
セッションキャッシュ	134
セッションチケット	146
ゼロ往復時間	165
選択的暗号文攻撃	42

前方秘匿性	36,165
<b>【そ】</b>	
素因数分解問題	40
操作トランザクション	55,83
属性	84
属性型	62
属性記述	85
属性値	62,85
属性値アサーション	85
ソルト	38,45,100,159,170,196
<b>【た】</b>	
帯域外手順	93
対称鍵暗号方式	3
楕円曲線暗号	6,49
楕円 DSA	51
楕円 Diffie-Hellman 鍵共有	51
<b>【ち】</b>	
チケット	146,168,185
中間 CA	56
中間 CA 証明書	57
中間者攻撃	36
抽象型オブジェクトクラス	85
抽象構文記法 1	88
次の発行日時	78
<b>【て】</b>	
ディレクトリ	84
ディレクトリエントリ	84
ディレクトリ情報木	85
ディレクトリ情報ベース	84
デジタル署名	8
デフレート圧縮	131
デルタ CRL	60,77
<b>【と】</b>	
登録	92
登録局	55
トークン	109,191
トラストアンカ	56
トランスポートパラメータ	197
トランスポートレイヤ セキュリティ	125
トリプル DES	18
<b>【な】</b>	
名前制約	68

名前付き曲線	50
名前付きグループ	146
ナンス	13,22,112,133,196
<b>【に】</b>	
認証	3,7
認証局	10,55
認証付き暗号	8,29
認証フェーズ	166
認証レベル	57
<b>【は】</b>	
パケット番号	190
バージョン	61,77
パスワード認証鍵共有	37
発行者	62,78
発行者代替名	68,80
発行日時	78
発行分配ポイント	80
ハッシュ関数	8
パディング	22
ハンドシェイクプロトコル	127
<b>【ひ】</b>	
非圧縮形式	76
非対称鍵暗号方式	6
秘密鍵	4,6
——の所有証明	94
秘密鍵暗号方式	3
標準エクステンション	70
平文	2
<b>【ふ】</b>	
復号	2
プライベートインターネット エクステンション	70
プレマスタシークレット	128
フル完全 CRL	77
フルハンドシェイク	133
フレーム	191
フレームタイプ	191
プレーン RSA	42
ブロック暗号	5
<b>【へ】</b>	
ベース CRL	77
ベースポイント	49
ヘッダ保護	196

<b>【ほ】</b>	
ポイズンエクステンション	72
補助型オブジェクトクラス	85
ポリシー制約	69
ポリシーマッピング	68
本人確認	93
<b>【ま】</b>	
マスタシークレット	128
<b>【む】</b>	
無効化日時	82
<b>【め】</b>	
メッセージダイジェスト	8
メッセージ認証コード	8,27
<b>【も】</b>	
モンゴメリ型楕円曲線	50
<b>【ゆ】</b>	
有効期間	63
ユーザ通知	67
<b>【よ】</b>	
予想される証明パス	122
<b>【り】</b>	
離散対数問題	34
リファレンス番号	96
理由コード	81
利用モード	5,21
<b>【る】</b>	
ルート CA	56
ルート証明書	57
<b>【れ】</b>	
レポジトリ	55
<b>【ろ】</b>	
ロールオーバー	111
ロングパケットヘッダ	190
<b>【わ】</b>	
ワイエルシュトラス型楕円曲線	49

<b>【その他】</b>	
ACK フレーム	191
AAD	30,31,133,196

AEAD	8,29,125,165
AEAD_CHACHA20_POLY1305	32
AES	5,18
ALPN 名	148

application_layer_protocol_ negotiation	148,177
ASN.1	88
authenticity	7



AVA	85	ElGamal 暗号	6,47	post_handshake_auth	181
availability	2	Ephemeral-Static モード	36	PRF	128
BASEKEY	100	EST	92,114	pre_shared_key	179
CA	55	EV 証明書	58	PSK	9,135
CA 証明書	57	extended_master_secret	144	psk_key_exchange_modes	150,179
CBC-MAC	28	GCM	31	QUIC	187
CBC モード	22	GeneralizedTime	63	quic_transport_parameters	197
CCM	30	GeneralName	68	RA	55
certificate_authorities	180	GMAC	32	RC4	5,11
ChaCha20	12	GREASE	143	renegotiation_info	145
CFB モード	22	Handshake パケット	190	RSA 暗号	6,40
Change Cipher Spec プロトコル	127	HelloRetryRequest クッキー	180	RSAES-OAEP	42
client_certificate_type	182	HMAC	27	RSAES-PKCS1-v1_5	44
client_certificate_url	151	HKDF	165	RSASSA-PKCS1-v1_5	46
CMAC	28	HKDF-Expand 関数	170	RSASSA-PSS	45,165
CMC	92,104	HKDF-Extract 関数	170	SCEP	92,111
CMP	92,96	HoL ブロッキング	189	SCT リスト	71
CMS	104	IAK	96	SCVP	120
——上の証明書管理	92	integrity	2	server_certificate_type	182
compress_certificate	150	Initial パケット	190	server_name	144
confidentiality	2	IV	22,172,196	session_ticket	146
cookies	180	key_share	149,176	SHA-1	25
CPS ポインタ	67	LDAP	84	SHA-2	26
CRMF	97	MAC	8,27	SHA-224	26
CRL	55	max_fragment_length	151	SHA-256	26
CRL 番号	80	MD5	24	SHA-384	26
CRL 分配ポイント	69	named curve	50	SHA-512	26
CRL エントリエクステンション	81	newWithNew	95	signature_algorithms	149,178
CRL エクステンション	79	newWithOld	95	signature_algorithms_cert	178
CRL 発行者	55	OCSP	56,116	signed_certificate_timestamp	149,183
CRL スコープ	77	OCSP クライアント	116	SRP	38,115,135,157
CRYPTO フレーム	192	OCSP レスポンド	116	SSL	125
CT	71	OCSP Stapling	117,148,183	Static-Static モード	36
CT ログ	71	OCSP status	183	status_request	148
CTR モード	22	OFB モード	22	STREAM フレーム	192
DES	5,15	OID	62	subjectPublicKeyInfo	63
DIB	84	oid_filters	180	supported_groups	146,177
Diffie-Hellman 鍵共有	34	oidWithNew	95	supported_versions	150,177
DIT	85	oldWithOld	95	TLS	125
DN	62	OOB 手順	93,96,109,115,169	TLS コネクション	128
DSA	48	OV 証明書	57	TLS ハンドシェイクプロトコル	133
DSS	48	padding	151	TLS レコードプロトコル	130
DPD	121	PADDING フレーム	191	TLS セッション	127
DPV	121	PAKE	37	TLS-SRP	157
Derive-Secret	171	PKCS	23	truncated_hmac	152
DTLS	159	PKCS #5 パディング	23	trusted_ca_keys	151
DV 証明書	57	PKCS #7 パディング	23	UTCTime	63
early_data	182	PKI	10,55	0-RTT	165
ECB モード	22	PKI 管理プロトコル	92	0-RTT パケット	190
ECC	49	PKI メッセージ	97	1-RTT	167
ECDH 鍵共有	51	PKI 操作プロトコル	83	1-RTT パケット	190
ECDSA	51	PKIX	55		
ec_point_formats	146	Poly1305	28		
EdDSA	53	POODLE	125		
		POP	94,101,109		



— 著者略歴 —

1978年 東京大学工学部電気工学科卒業  
1983年 東京大学大学院工学系研究科博士課程修了（電子工学専門課程）  
工学博士  
1983年 国際電信電話株式会社研究所勤務  
1991年 国際電信電話株式会社研究所主任研究員  
1995年 国際電信電話株式会社研究所グループリーダー  
2001年 KDDI 研究所執行役員  
2002年 電気通信大学助教授  
2007年 電気通信大学教授  
2021年 電気通信大学名誉教授  
2022年 神奈川工科大学客員教授  
現在に至る

ネットワークセキュリティ詳説 **PKI/TLS プロトコル**

PKI / TLS Protocols : Network Security Explicated

© Toshihiko Kato 2022

2022年10月3日 初版第1刷発行

★

検印省略

著者 加藤 聡彦  
発行者 株式会社 コロナ社  
代表者 牛来真也  
印刷所 壮光舎印刷株式会社  
製本所 株式会社 グリーン

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替00140-8-14844・電話(03)3941-3131(代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-02929-1 C3055 Printed in Japan

(西村)



 < 出版者著作権管理機構 委託出版物 >

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつと事前に、出版者著作権管理機構（電話 03-5244-5088, FAX 03-5244-5089, e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたしません。