**## 1.1 SSLからTLSへ**

**### デファクト標準SSL**

SSLは当時急速に普及しはじめていたWebブラウザの中心的ベンダーであったネットスケープコミュニケーションズ社によって開発され同社のブラウザーに実装されました。初期のバージョンには重要な脆弱性が発見されましたが、1996年に提供されたSSL3.0は安全面でも当時としては大幅に改善されインターネット黎明期のデファクト標準としてその後広く利用されるようになりました。

**### 中立的標準TLS**

一方、インターネットの急激な普及に伴ってセキュリティの重要性が認識されるようになり、特定企業によらないセキュリティプロトコル標準の必要性に対する認識が高まってきました。それと同時に標準策定組織としてIETFの基盤もかたまりつつあり、1996年にはIETFによるTLSの仕様策定が開始されました。しかし、最初のバージョンであるTLS1.0は中立的な組織による標準プロトコルとしての意味はあったもの、当時すでに広く普及していたSSL3.0と大きな違いがない一方互換性はなく広く普及するには至りませんでした。

**### 標準の成熟**

インターネットの世界ではその後も次々に新たな攻撃方法、プロトコル仕様レベルの脆弱性リスクが指摘され、TLSにもIETFでの改定努力が継続されました。2006年にはTLS1.1(RFC-4346)、2008年にはTLS1.2(RFC-5246)が発行されるまでの過程で、セキュリティプロトコル仕様に必要な要件も次第に明らかになりプロトコルの堅牢性も高まっていきました。またその中で、常に進化する攻撃手法、セキュリティ、暗号技術に柔軟に対応していくための仕組みも確立し、TLSはインターネットセキュリティの中核をになう標準プロトコルとして必須のものとなるに至りました。

**### 新たなTLSの検討**

しかしその一方、進化の過程で後方互換性を維持することによる複雑さの増大、新たな潜在的な脆弱性リスクも危惧されるようになり、2013年にはこれまでの経験と実績をふまえた集大成としての新しいTLS2.0としてメジャーバージョンに向けた検討が開始されました。しかし、TLSのバージョン情報は実際のネットワークを行きかうTLSレコード中に埋め込まれているために、スムーズにメジャーバージョンを上げることは極めて難しいことが検討開始ほどなく判明。検討目標はそのまま堅持するものの、バージョンの呼称としてはやむなくTLS1.3とマイナーバージョンアップの呼称とすることになりました。

そのような経緯でバージョンの呼称は控え目なものとなっていますが、最終的に2018年に標準として発行されたプロトコル仕様は当初の目標通りにこれまでの懸案を一掃するメジャーバージョンアップに匹敵する仕様改定となりました。

**### 新しい脅威**

そんな中、今まで知られていなかったような大規模な組織的なネットワーク通信の傍受、情報収集の事実が知られ、完全前方秘匿性のような新たな次元のセキュリティの必要性が認識されるようになりました。そのため今まで安全とされていた公開鍵暗号についても、その使用方法を見直す必要性が指摘されるようになりました。

**### 新たな暗号技術**

その間、新しい暗号アルゴリズムの研究も進み、安全性とともに従来よりも格段に処理効率のよいアルゴリズムが実用的に利用できるようになってきました。そうした個別のアルゴリズムはTLS1.2の中でも標準として順次取り入れられ、古い危殆化したアルゴリズムも徐々に廃止されてきましたが、後方互換性のためにリスクのあるアルゴリズムの一掃には至っていませんでした。また、プロトコルや暗号アルゴリズムに対する形式的検証や検証ツールについても現実の検証に有効なものが登場しはじめていました。

**### TLS1.3**

こうした背景から、2018年8月に正式発行されたTLS1.3ではこれまでの後方互換性を捨て、プロトコル仕様の大胆な整理が実現されました。もちろん、現実に日々使用される世界中のプロトコルを一気にバージョンアップすることは不可能です。実装上は旧バージョンのTLSも受け入れるようにしつつ新しいバージョンへの移行が行えるようになっています。しかし、新しいTLS1.3同士で確立した通信セッションに関しては、セキュリティリスクが最大限排除され、性能面からも改善が期待されるものとなっています。以下にTLS1.3の安全性と性能面での改善点についていくつかあげてみます。

**### 安全性の改善**

- ハンドシェークの大部分を暗号化

- 暗号化アルゴリズムを整理

- ダウングレード、圧縮など危険な機能の廃止

- 静的公開鍵を排除、一時鍵ディフィーヘルマン系のみに絞り込み

- TLSレコードのMACを廃止し、AEAD共通鍵暗号のみに絞り込み

これらに伴い、暗号スイートを大幅に絞り込み、暗号化スイート表記法を整理

**### 性能の改善**

- ハンドシェークの整理による往復数の削減、レイテンシーの低減

- 処理効率の高い新しい暗号アルゴリズム、楕円曲線の正式標準化

**### 進化**

SSLとTLSの両者はクライアントとサーバーで行われるハンドシェークを指す言葉としては今だ同義語として使われ続けています。しかし、プロトコル仕様を指す言葉としてのSSLは今や過去の仕様であり実際の製品では使われることはありません。TLSはさらにその後も進化を継続しTLS1.3で一応の集大成を実現しました。本書では、そうした認識に基づき特別の断りがないかぎりTLS1.3にもとづいて解説します。

TLS1.3でプロトコル仕様としてまずは完成度の高いものが標準化されましたが、システム全体として安全なネットワーク通信を実現するためには、安全なプロトコルを実現するライブラリー層での高い品質の実現、それを使用するアプリケーション側でのセキュリティとプロトコルに関する正しい理解や使用方法、またそのアプリケーション、システムの運用方法などすべてのレイヤーで確実なセキュリティを実現する必要があります。

しかし、アプリケーションエンジニアがTLSライブラリーを正しく理解して安全なシステムを実現するために必要となる知識は多岐にわたり、これまでそれぞれの専門分野、テクノロジーのレイヤーなどに分散しがちでした。本書ではそうした知識を一貫した形で理解できるように整理し、説明していきます。