暗号プロトコル安全性評価に関する
調査報告書

2003 年 1 月 31 日

株式会社ケーディーディーアイ研究所

暗号プロトコル安全性評価に関する
調査報告書

平成15年1月31日
株式会社KDDI研究所

目 次

1　本調査の目的

本調査は、昨年度実施したSSLプロトコルの安全性評価の結果を踏まえ、電子政府システムで今後適用される可能性がある他の暗号プロトコルの候補をリストアップするとともに、これら一般的な暗号プロトコルに対する安全性評価の検討を行う際のガイドラインを提示することを目的としている。

1.1　背景

暗号アルゴリズムの安全性評価を行う技術については、昨今、格段に進歩を遂げてきており、線形解読や差分解読などの各種攻撃に対する暗号アルゴリズムの耐性に関する指標や、証明可能安全性といった技術が発達しており、客観的な評価を行うに値する十分な指標が明確となっている。一方、暗号プロトコルは、２つ以上のエンティティ間で、暗号アルゴリズムをベースに認証や鍵共有といったセキュリティ機能を提供する手順であり、セキュリティ機能や利用要件に従って、さまざまな方式が提案されてきた。しかしながら、これらの暗号プロトコルの安全性について、厳密に評価する手法が確立されておらず、プロトコル設計はさまざまな攻撃を回避するためにヒューリスティックな精査を行い、仕様を確定する必要がある。従って、その過程において仕様検討の不十分から、仕様確定後やその製品が提供されてから、そのプロトコルの問題が指摘される場合が生じている。昨今、ある暗号プリミティブ（暗号アルゴリズム）が安全であるという仮定のもとに、その暗号アルゴリズムを用いた暗号プロトコルが安全であることを証明できる証明可能安全性付きプロトコルも一部、提案されているが、手順の複雑なプロトコルでは証明が困難である点から既存プロトコルを評価するには、解決すべき課題がある。

一方、電子政府においては、PKIを用いた電子認証システムが稼動し始めており、そのPKI基盤を用いた各種暗号プロトコルが、アプリケーションで必要なセキュリティ機能を実現する手段として用いられ始めている。今後、電子政府の発展にともない、様々なアプリケーションやそのアプリケーションで必要となる各種暗号プロトコルが利用される可能性がある。しかしながら、これら暗号プロトコルの安全性について、一定の基準を満足するかどうかの判断が現状困難であり、一部のプロトコルの脆弱性によりシステム全体が破綻してしまう可能性もある。

このような課題を解決するために、暗号プロトコルの安全性について、客観的な評価ができる技術を確立することは、今後の電子政府の高度化、サービスの多様化を実現するためには不可欠であると考えられる。上記の背景を踏まえ、本調査は、暗号プロトコルの安全性評価に関して、現在利用されている、あるいは、今後、導入される可能性のある標準的な暗号プロトコルの候補をリストアップし、暗号プロトコル技術のロードマップとして提示するとともに、昨年度実施したSSLプロトコルの安全性評価の結果を踏まえて、

他のプロトコルに適用可能な安全性評価を客観的に行うための指標について調査検討を行う。

1.2 調査方法

本調査を行なうにあたっては、まず、現状検討が進められている電子政府システムについて、調査を行ない、どのような暗号プロトコルが利用されているかの洗出しを行なう。その結果として、電子政府で用いられている暗号プロトコルのリストを作成する。

リストであげた暗号プロトコルを対象として、現在知られている脆弱性について、仕様、実装、運用の観点から調査を行なう。これらの調査と、現状の技術動向から、今後電子政府として利用が予想される暗号プロトコルの技術マップを検討する。

最後に、調査の結果得られた暗号プロトコルの脆弱性を体系的に整理し、それらの脆弱性を回避するために考慮すべき点について、提言としてまとめる。

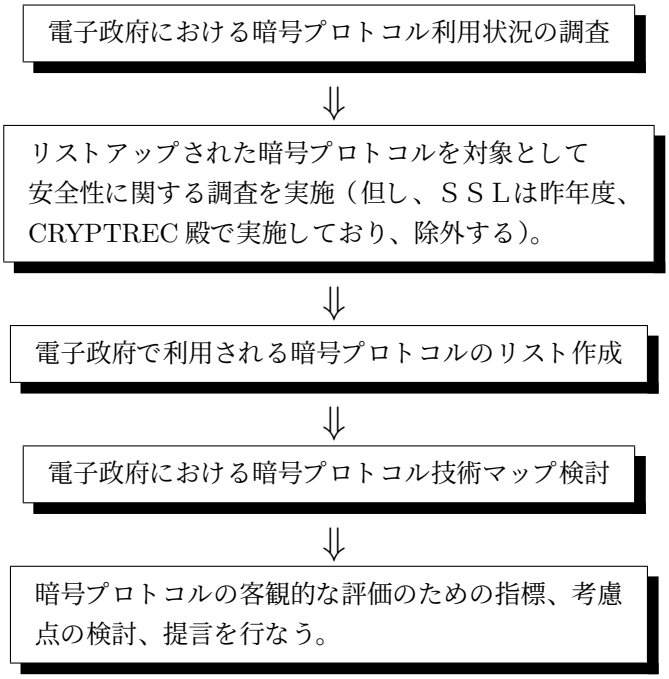


図 1: 調査作業の概要、流れ

2　電子政府システムの現状調査

電子政府は、行政情報の電子的提供や、政府の申請などの諸手続きを電子化するパブリックアクセスの電子化、歳入歳出、ペーパレス化、電子調達などを実現するパブリックマネジメントの電子化などを目的として、一部すでに実用化されているシステムや、現在導入に向け、設計開発を進めているシステムがある。電子政府システムの利用促進をはかるためには、これらのシステムを安全に運用することが重要な要件である。このため、個々のシステムのセキュリティ要件に合致したセキュリティ機能が実現されている。特に、電子政府システムを情報通信システムとしたとき、各種業務サーバやクライアント間でのトランザクションを安全に行なうための通信手順（プロトコル）は十分に安全性を配慮して設計する必要がある。これは、電子政府がCtoG、GtoC を目的とするオープンなシステムであることを考慮すると必須の課題であるといえる。本書では、このような各種セキュリティ機能を実現する通信手順を暗号プロトコルと呼ぶこととする。暗号プロトコルは、表１で示すように、鍵共有、相手認証、情報秘匿、データ完全性、否認防止などの基本機能に加えて、その他、オークションや、電子投票といったアプリケーションを直接実現するプロトコルを含む場合もある。本調査では、電子政府システムを実現するための、汎用的なプロトコルという位置付けで調査を行うことを目的とし、上記のアプリケーション固有の暗号プロトコルは調査の対象外とする。

本節では、公開された情報あるいは一部のヒアリングを通じて現状の電子政府システムで求められるセキュリティ要件および各システムで用いられる暗号プロトコルについて、調査を行った。

セキュリティ機能	**目的**
鍵共有	対話を通じて、通信エンティティ間で、後に利用する一時的な鍵を共有するセキュリティ機能
相手認証	対話を通じて、通信相手が、検証者の意図したエンティティであることを確認し、なりすましを防止するセキュリティ機能
情報秘匿	通信エンティティ間で、送信データを暗号化することにより、ネットワークを不正にモニターする盗聴を防止するセキュリティ機能
データ完全性	通信エンティティ間で、送信データを攻撃者が不正に改ざんするのを防止するセキュリティ機能
否認防止	対話を通じて、通信を行ったことを事後に否定するのを防止するセキュリティ機能

表 1: 暗号プロトコルで実現されるセキュリティ機能

2.1　電子政府システムアクションプラン

表 2, 表 3, 表 4 に、電子政府システム、地方自治体の電子行政システムのスケジュールを示す。各システムとも、主として、２００３年度から２００４年度を最終的な実現時期としている。

分野	事業項目	記載計画	概　要	2000年度	2001年度	2002年度	2003年度	2004年度	担当府省
パブリックアクセス	1．行政情報の電子的提供								
	行政情報の電子的提供に関する基本的考え方（指針）発表	重点計画	指針では、行政情報の原則ホームページによる提供、所在情報の充実、一元的提供、タイムリーな提供などを提示	●2001.3					情報化推進各省庁連絡会議
	実施方針策定、重点取り組み	重点計画	上記指針に即して各府省の実施方針策定、重点取り組み		●年度早期 重点取り組み期間				全府省
	2．申請・届出等手続きの電子化								
	国の手続きのオンライン化実施	重点計画 申請ｱｸｼｮﾝﾌﾟﾗﾝ	実質的に全ての申請・届出等手続きをインターネット等でオンライン化	できるだけ早期に		(35%をｵﾝﾗｲﾝ化)	(98%をｵﾝﾗｲﾝ化)		全府省
	アクションプラン見直し	重点計画	実施時期の前倒し、手続きそのものの抜本的見直し、事務処理の電子化の観点で見直し		●年度早期				総務省及び全府省
	オンライン化に伴う法令の見直し	重点計画	見直しの基本方針とりまとめ、法令の見直し		●方針 見直し				全府省
	共通的基盤システム整備	重点計画	府省認証システム、汎用受付等システムの整備						全府省
	汎用受付等システム基本仕様策定	重点計画	府省間で整合性を図る必要のあるものについて基本仕様をとりまとめ		●年度早期				総務省及び全府省
	各府省個別手続きのオンラインシステム整備	重点計画	各府省の個別手続きを可能な限り平成14年度中にオンライン化		可能な限り				全府省
	申請・届出等の総合窓口システム運用	重点計画	各府省が提供する案内情報や申請書様式についてホームページに掲載したものを横断的に検索できる総合窓口システム運用		●運用開始				総務省
	自治事務等のオンライン化アクションプラン策定	重点計画 ｱｸｼｮﾝﾌﾟﾗﾝ 申請ｱｸｼｮﾝﾌﾟﾗﾝ	地方公共団体からの要望等を踏まえて個別手続きにかかる標準仕様等の提示や法令改正等の時期についてアクションプランを策定		●年度早期	(実施方策59%提示)	(実施方策95%提示)		総務省及び関係府省
	地方公共団体汎用システム基本仕様策定	重点計画 ｱｸｼｮﾝﾌﾟﾗﾝ	地方公共団体の申請・届出の受付など複数の手続きに汎用的に利用できるシステムの基本仕様を策定	概要、スケジュール決定	●基本仕様決定				総務省及び関係府省
	電子自治体推進パイロット事業の推進	ｱｸｼｮﾝﾌﾟﾗﾝ	複数の自治体で電子申請、情報提供等を行うモデルシステムを構築、全国に普及						総務省
	地方税申告手続きオンライン化	ｱｸｼｮﾝﾌﾟﾗﾝ	地方税の申告手続きの電子化	推進協議会立ち上げ	研究開発、法令改正		●運用開始		総務省及び関係府省
	地方公共団体における組織認証基盤（LGPKI）の整備	ｱｸｼｮﾝﾌﾟﾗﾝ	地方公共団体が発信した電子文書が真にその団体によってなされたものかを確認するための基盤として構築	●基本仕様作成	●都道府県、政令指定都市構築		●未構築市町村構築推進		総務省
	地方公共団体における個人認証基盤の整備	ｱｸｼｮﾝﾌﾟﾗﾝ	申請者が発信した電子文書が真にその人によってなされたものかを確認するための基盤として構	制度的枠組み検討、実証実験実施			●運用開始		総務省

（注）重点計画：e-JAPAN重点計画　ｱｸｼｮﾝﾌﾟﾗﾝ：地域IT推進のための自治省ｱｸｼｮﾝﾌﾟﾗﾝ　申請ｱｸｼｮﾝﾌﾟﾗﾝ：申請・届出等手続のｵﾝﾗｲﾝ化に係るｱｸｼｮﾝﾌﾟﾗﾝ

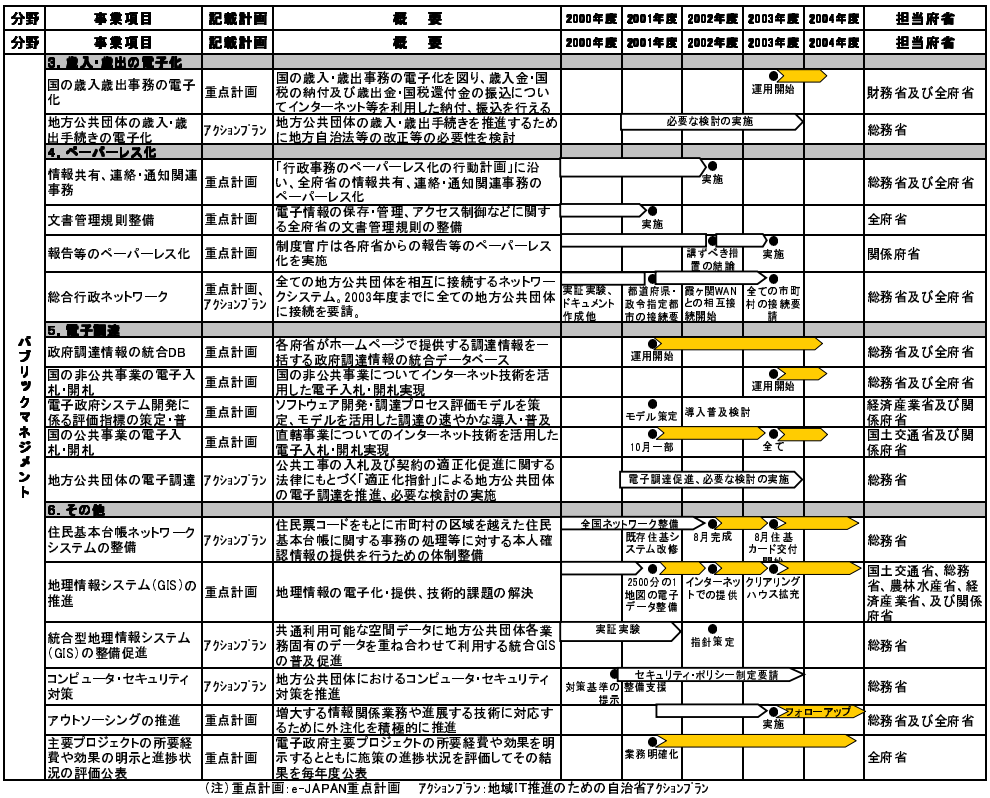
表 2: 電子自治体関連事業の概要とスケジュール (1)

2.2　電子政府システムにおける暗号プロトコルの実態調査

公開情報および一部ヒアリングから得られた電子政府システム関連の概要を別紙１に示す。ここでは、これらのなかから、いくつかの典型的な電子政府システムに関して、重点的に、暗号プロトコルの利用状況について調査を行った。

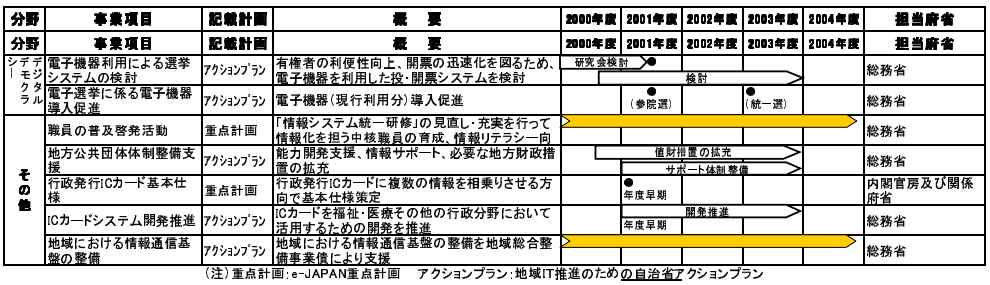
(1) 電子申請システム

電子申請システムは、各省庁、自治体への導入が最も先行しているアプリケーションである。従って、比較的多くの情報が収集可能であった。ここでは、特に、経済産業省が開発した汎用電子申請システム ITEM2000 について、詳細な調査を行い整理表としてまとめている (表 5)。公文書の申請受け付け、および、公文書の申請者への発行処理は、事前配布されたプログラムを用いた独自プロトコルが用いられている。一部、証明書取得の手法として、ＳＳＬが用いられている。



（注）重点計画：e-JAPAN重点計画　アクションプラン：地域IT推進のための自治省アクションプラン

表 3: 電子自治体関連事業の概要とスケジュール (2)



（注）重点計画：e-JAPAN重点計画　アクションプラン：地域IT推進のための自治省アクションプラン

表 4: 電子自治体関連事業の概要とスケジュール (3)

(2) 電子調達システム
電子調達システムについては、国土交通省が早期から着手しているシステムであり、近年では自治体への導入も進んできている。そのため、公開ベースで比較的詳細な資料が入手可能である。ここでは、国土交通省の CALS/EC システムを自治体へ普及するために、一部機能（PPI, 入札）をライブラリ化した「コアシステム」および、現在、各都道府県で検討が進んでいる自治体の共同運営システム（兵庫県の場合を例にとる）の２種類について、整理表としてまとめている (表 6, 表 7)。コアシステムについては、入札者の登録・参加資格確認、入札処理・開札処理について、ＳＳＬが用いられている。一部、参加資格確認申請受け付け票の送付について S/MIME が用いられている。地方自治体の共同運営システムについても、各種申請処理についてはＳＳＬが、申請や入札結果などの参加者への通知に S/MIME が用いられている。

(3) 債権譲渡登記オンライン申請システム
電子申請システムの他の形態として、債権譲渡登記オンライン申請システムについても整理表としてまとめている (表 8)。本システムについては、登記申請受け付けおよび、証明書発行処理として、事前配布プログラムを用いた独自プロトコルを用いている。

(4) 国税電子申請・納税システム
本システムは納税者や税理士などが電子申告を行うシステムである。本システムについては、特に整理表としてまとめていないが、申告書のデータ形式に XML を用いている点、セキュリティ要件として電子署名が必要となる点から、特徴的なシステムとなっている。

これらの具体的なアプリケーションとは、別に電子政府の基盤技術と考えられる住民基本台帳カードや政府認証基盤、さらに、電子政府アプリケーションが稼動するバックボーンネットワークという位置付けで、総合行政ネットワーク（いわゆるＬGWAN）をについても、暗号プロトコルの観点から調査を行った。また、電子政府システムとしては、若干他のシステムと異なるが、料金の支払いは、各種申請において必要となるため、決済システムが検討されている。これらのシステムについての調査結果の概要を以下に示す。

(5) 住民基本台帳カード
公的 IC カードの１つとして、最も注目されている。昨年（H14）に「公的分野における連携 IC カード技術仕様」として公開された。IC カードとしては、JIS X 6322 いわゆる ISO/ IEC 14443 に相当する非接触（近接型）IC カード標準への準拠をうたっている。ISO/IEC 14443-4 では、伝送プロトコルとして、セキュリティ通信手順が規定されている。

(6) 政府認証基盤（GPKI）
電子申請、電子調達など、GtoC, CtoG のセキュリティを確保するため

の最も重要なセキュリティ要素技術であり、IC カードとの連携も深い基盤技術である。調査においては、

- 政府認証基盤（GPKI）国土交通省認証局証明書ポリシー・認証実施規定
- 政府認証基盤（GPKI）府省認証局 CP/CPS ガイドライン

などを参考にした。これらは、いわゆる PKI 技術に準拠した仕様となっている。PKI 技術としては、暗号プロトコルの範疇に入りがたいが、一方で、ＰＫＣ (Public Key Certificate) やＣＲＬ (Certificate Revocation List) の運用について証明書管理プロトコルが規定されている。しかしながら、現状の政府認証基盤では、利用についてのこれらの管理プロトコルに対する詳細な規定がない。

(7) 総合行政ネットワーク

総合行政ネットワーク（ＬGWAN）は、地方自治体における電子政府の基盤と位置付けられている。ＬGWANは、地方公共団体の組織内ネットワークを相互に接続し、高度情報流通を可能とする通信ネットワークとして整備し、地方公共団体相互のコミュニケーションの円滑化、情報の共有による情報の高度利用等を図ることにより、各地方公共団体と国の各省庁及び住民等との間の情報交換手段の確保のための基盤とすることを目的とする。本調査においては、

・総合行政ネットワークに関する実証実験報告書

を参考にした。本調査では、実証実験について行なっており、実運用との差異を明確に調査していない点に留意されたい。ただし、この実験の仕様には全国自治体への拡張性が謳われていること、多くの自治体 (58 団体) が参加していること、実験の結果から技術面で大きな課題があげられていないことから、本格運用に移行しつつある現在の LGWAN の仕様も大きな変更はないものと推定される。ＬGWANは、別紙に示すとおり、県域アクセス回線には、IP-VPN/MPLS を、ネットワークの暗号プロトコルとして IPsec(暗号化方式は T-DES) を、ＩＫＥ認証として、証明書のフォーマット（ITU-TX.509 V.3）を、証明書検証サーバのアクセスプロトコルとして、OCSP を利用している。

(8) マルチペイメントネットワーク

マルチペイメントネットワークとは、各種企業・団体と金融機関を結び、利用者と企業・団体間に発生する各種の決済にかかわるデータを転送するためのネットワークをさす。マルチペイメントネットワークは、別紙に示すとおり、閉域網を想定しており、IKE を用いた相手認証や IPsec を用いた暗号通信路を確保している。

表 5: 汎用電子申請システム ITEM2000

資料 ･･･ 汎用電子申請システム＜ ･･･････＞

対象システム	所管省庁	系	処理手順 官 ･ 大分類	処理手順 官 ･ 小分類	媒体	送受	媒体	処理手順 民 ･ 小分類	処理手順 民 ･ 大分類	系	分類 ･ ●：現状 ○：将来	分類 ･	分類 ･ 秘匿性	分類 ･ 保存	セキュア仕様 暗号プロトコルの種類	暗号技術の種類・仕様	鍵の配布方法	署名の有無 ･方式	･･利用の有無	･･カード利用の有無
汎用電子申請システム ･･･････	経済産業省 ※注 厚生労働省も ･･･ ベースのシステムを導入している	業務	準備	申請用ソフトウェア配布	配付：･･･		配付：･･･	申請用ソフトウェア入手	準備											
								端末 ･･･ へインストール												
								意口・インターネット→認証局 ･			●･･･ ●･･･ ○国民 ･基盤	相手認証 鍵共有 守秘 署名	高	法令などにより定められる期間 ･	･･･	鍵配信 ･	電子証明書 ･ ･･カード ･	有	有 ･･	有 ･･
								証明書取得												
								端末 ･･･ への組込						証明期間 ･						
				審査官情報の登録	･･･															
				処分権者の官職証明書取得							･･･	守秘	高	法令などにより定められる期間 ･	･･･独自 ･	鍵配信 ･	･安全且つ確実な方法 ･ ･電子証明書 ･可搬メディア ･ ･･カード	有	有 ･･	有 ･･
				担当者振り分け設定																
				申請様式の登録																
		受付	受付	申請様式の電子配布	インターネット	→	インターネット	申請様式取得	受付		●･･･ ●･･･	相手認証	低	･改定時に更新 ･	独自 ･監事プログラムで提供 ･					
								申請書作成												
								申請書署名				守秘 署名	高			･･･ハッシュ：･･･ ･	電子証明書	有	有 ･･	有 ･･
				申請書受信 ･申請書パッケージ受信 ･	インターネット	←	インターネット	申請書送信 ･申請書パッケージ送信 ･			●･･･ ●･･･ ○国民 ･基盤 ○企業 ･基盤	相手認証 鍵共有 守秘 署名	高	法令などにより定められる期間 ･	独自：電子申請通信プロトコル ･監事プログラムで提供 ･	鍵配信 ･ ･･･ハッシュ：･･･ ･	電子証明書 ･ ･･カード ･ に記録	有	有 ･･	有 ･･
			審査・決裁	認証・検証	･インターネット→認証局 ･				審査・決裁						･･･独自 ･	鍵配信 ･	電子証明書	有	有 ･･	有 ･･
				一次形式審査	･･･											･･･ハッシュ：･･･ ･	電子証明書			
				到達申請書登録																
				原課担当者への振り分け																
				原課担当者による審査 ･二次形式審査 ･目視による内容審査 ･	インターネット	→	インターネット	申請書補正			●･･･ ●･･･	相手認証 守秘	高		独自：電子申請通信プロトコル ･監事プログラムで提供 ･	鍵配信 ･	電子証明書	有	有 ･･	有 ･･
				※必要な場合は申	インターネット	←	インターネット	再送信			●･･･ ●･･･ ○国民 ･基盤 ○企業 ･基盤	相手認証 鍵共有 守秘 署名	高			･･･ハッシュ：･･･ ･	電子証明書 ･ ･･カード ･ に記録	有	有 ･･	有 ･･
				認証・検証	･インターネット→認証局 ･										･･･独自 ･	鍵配信 ･	電子証明書	有	有 ･･	有 ･･
			公文書発行	決裁	･･･				公文書発行		･･･	守秘	高							
				公文書作成																
				官職証明書添付								守秘 署名								
				公文書発行	インターネット	→	インターネット	公文書取得			●･･･ ●･･･ ○国民 ･基盤 ○企業 ･基盤	相手認証 鍵共有 守秘 署名	高		独自：電子申請通信プロトコル ･監事プログラムで提供 ･	･･･ハッシュ：･･･ ･	電子証明書	有	有 ･･	有 ･･
								公文書確認 ･官職証明書検証 ･								鍵配信 ･ ･･･ハッシュ：･･･ ･	秘密鍵 ･･･カードに記録 ･	有	有 ･･	有 ･･
				原本性保証保存	･･･						･･･	守秘	高	法令などにより定められる期間 ･						

･申請者と汎用受付等システム間通信で利用されるプロトコル ･

・電子申請業務プロトコル：申請・届出等手続に関するデータ送受信用の上位プロトコル。
申請書様式、添付様式、公文書等を構成する「申請書パッケージ」と電子申請の手続状況を管理する「手続構成ファイル」を ･･形式でアーカイブしたファイルをセキュア通信プロトコルへ引き渡し及び引き受けを行う。
電子様式の取得、提出、電子公文書の取得、審査状況の確認等を行う。

・電子申請通信プロトコル：通信経路上のセキュリティを確保し、効率良くデータを送受信するための下位プロトコル。
･･･対応の電子証明書で本人確認、電子様式のバージョン確認、データ容量の確認、通信予約の制御等を行う。
暗号化、データ分割、通信障害時の再送制御、送達確認、時刻記録等を行う。
暗号化及びデータ分割して送られた申請書パッケージを復元し、電子署名を ･･･で検証し、本人確認及び改ざん検知を行う。

･･･ ･･･････ における ･･･と ･･カードの利用 ･

・官職の認証には経済産業省の政府認証基盤 ･･･ ･･ を利用している。
・ ･･･ の仕様では、官職証明書の秘密鍵は官職者携帯の ･･カードにより保護することとしている。 ･詳細は資料「政府認証基盤 ･･･ ･府省認証局 ･･･ ･ガイドライン」をご参照 ･
・申請者の認証は、「 ･･･ のブリッジ認証局との間で相互認証する認証局から発行された申請者の証明書が必要」とされており、 ･･･ のブリッジ認証局との間で相互認証する認証局とは、「商業登記認証局」、特定認証業務認定事業者が運営する認証サービス ･平成 ･年 ･月現在で ･種 ･などが挙げられる。
・なお、特定認証業務認定事業者が運営する認証サービスの中には、当該サービスを利用するための ･･カードを発行しているものもある。
・また、 ･･･ の仕組みの中で利用されている官職者が携帯する ･･カードは、将来的に住民基本台帳カードと同一の仕様のカードとなる可能性がある。

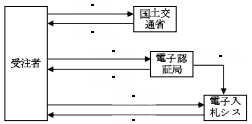
表 6: コアシステム

資料 電子入札システム<コアシステム 工事案件入札機能のみ >

対象システム	所管省庁	系	処理手順 発注者 大分類	処理手順 発注者 小分類	媒体	送受	媒体	処理手順 受注者 小分類	処理手順 受注者 大分類	系	分類 ●：現状 ○：将来	分類	分類 秘匿性	分類 保存	セキュア仕様 暗号プロトコルの種類	暗号技術の種類・仕様	鍵の配布方法	署名の有無・方式	利用の有無	カード利用の有無
電子入札システム コアシステム ※注 国土交通省の 電子入札関連機能を自治体がカスタマイズして利用できるようにライブラリ化したものの セキュリティ関連の仕様は国土交通省の も同様。	自治体	コアシステム	提供	カード 電子証明書・暗証番号 ※カードには、ゼロトの公開鍵等の鍵ペア、証書、秘密鍵が格納されて	窓口・郵送	→	窓口・郵送	カード 電子証明書・暗証番号入手	準備	コアシステム	● ● ○国民 盤 ○企業 盤	相手認証鍵共有 守秘 署名	高	法令などにより定められる期間						
				電子入札専用ソフトウェア	配付：	→	配付：	電子入札専用ソフトウェア入手												
								端末 インストール												
			登録・入札参加資格確認	申請受付	インターネット	←	インターネット	利用者登録申請	登録・入札参加資格確認		● ● ○国民 盤	相手認証鍵共有 守秘 署名	高	法令などにより定められる期間		使用する共通鍵暗号と鍵サイズは、 間	電子証明書	有	有	有
				申請書検証	インターネット 認証局												電子証明書	有	有	無
				利用者登録 利用者																
				職員登録 利用者																
				調達案件の登録	インターネット	⇔	インターネット	案件検索 参照			● ●		低					無	有	無
				参加資格確認申請受付	インターネット	←	インターネット	参加資格確認申請送信				相手認証鍵共有 守秘 署名	高	法令などにより定められる期間			電子証明書	有	有	有
				参加資格確認申請受付票	インターネット	→	インターネット	参加資格確認申請受付票受取										有	有	有
				署名検証	インターネット 認証局													有	有	無
							インターネット 認証局	署名検証										有	有	有
				参加資格確認通知書発行	インターネット	→	インターネット	参加資格確認通知書受取									入札用公開鍵で暗号化	電子証明書	有	有	有
							インターネット 認証局	署名検証										有	有	有
			入札・開札	入札書・内訳書受付	インターネット	←	インターネット	入札書・内訳書送信	入札・開札		● ●	相手認証鍵共有 守秘 署名	高	法令などにより定められる期間		共通鍵で暗号化	電子証明書	有	有	有
				入札書・内訳書受付票発行	インターネット	→	インターネット	入札書・内訳書受付票受取										有	有	無
							インターネット 認証局	署名検証										有	有	有
				入札締切通知書発行	インターネット	→	インターネット	入札締切通知書受取										有	有	無
							インターネット 認証局	署名検証										有	有	有
				入札書・内訳書共通鍵要求発行	インターネット	→	インターネット	入札書・内訳書共通鍵要求受取										有	有	無
							インターネット 認証局	署名検証										有	有	有
				入札書・内訳書共通鍵受付	インターネット	←	インターネット	入札書・内訳書共通鍵送信									入札用公開鍵で暗号化	電子証明書	有	有	有
				入札書・内訳書共通鍵受付票発行	インターネット	→	インターネット	入札書・内訳書共通鍵受付票受取										有	有	無
							インターネット 認証局	署名検証										有	有	有
				開札																
				共通鍵を秘密鍵で復号																
				入札書・内訳書を共通鍵で復号																
				落札者決定通知発行	インターネット	→	インターネット	落札者決定通知受取			● ●	相手認証鍵共有 守秘 署名	高				電子証明書	有	有	無
							インターネット 認証局	署名検証										有	有	有
			結果公開	入札結果を公開	インターネッ	⇔	インターネッ	閲覧	結果公開				低							
			契約						契約											
			支払						受領											

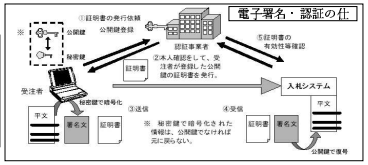
※他システムとの連携では、「資格審査情報」「調達情報」「契約・財務情報」を取得する。外部とのデータ送受ではセキュア通信を利用する。

電子入札システム全般の手続



・国土交通省に対して一般競争参加資格審査申請を行う。
・国土交通省から一般競争 指名競争 参加資格認定通知書・資格決定通知書が発行される。
・電子認証局に電子認証書発行を申請する。
・電子認証書 カードに格納 を発行。
・受注者が電子入札システムへアクセスするために必要な電子認証書発行情報を電子入札システムに提供する。
・電子認証書 カードに格納 を利用して電子入札に参加

電子署名・認証の仕



・ ・コアシステムにおける と カードの利用 ・官職の認証には地方公共団体組織認証基盤 を利用している。
・ への仕様は政府認証基盤 と共通部分が多く、官職証明書の秘密鍵は官職者携帯の カードにより保護することとしている。 詳細は資料「政府認証基盤 府省認証局 ガイドライン」をご参照 ・申請者の認証は、特定認証業務認定事業者である、帝国データバンク、日本電子認証、東北インフォメーション・システムズ、サイバートゥエイブジャパンなどが運営する認証サービスを利用する。
・上記 社が提供する認証サービスは、当該サービスを利用するための カードを発行している。

資料－－－ 電子入札・調達システム

対象システム	所管省庁	系	処理手順－官－ 大分類	処理手順－官－ 小分類	媒体	送受	媒体	処理手順－民；業者－ 小分類	処理手順－民；業者－ 大分類	系	分類－●：現状 ○：将来	分類－	分類－ 秘匿性	分類－ 保存	セキュア仕様 暗号プロトコルの種類	暗号技術の種類・仕様	鍵の配布方法	署名の有無－方式	利用の有無	カード利用の有無
電子入札・調達システム	兵庫県自治体共同運営システム		入札参加資格者登録	入札参加資格申請受付	インターネット	←	インターネット	入札参加資格申請	入札参加資格登録申請		●－－－－ ●－－－－ ○国民－基盤 ○企業－基盤	相手認証 鍵共有 守秘 署名	高	－法令などにより定められる期間－	－－－	－署名－ －－－－－－－－－－ ハッシュ： －－－－－ －－	電子証明書 －ＩＣカードに記録－	有	有	有
				申請書検証	－インターネット→認証局－										－－－－独自－	－－	電子証明書	有	有	無
				入札参加資格審査 審査情報収集 入札参加資格者登録	－－－						－－－	守秘	高							
				入札参加資格審査完了通知	インターネット	→	インターネット	通知書受取	結果受取		●－－－－ ●－－－－ ○国民－基盤 ○企業－基盤	相手認証 鍵共有 守秘 署名	高		－－－－－－	－－	電子証明書	有	有	無
								通知書確認									－電子証明書－ＩＣカードに記録－	有	有	有
			入札・調達情報の提供	入札・調達案件登録	－－－				入札・調達情報の入手		－－－		低	－法令などにより定められる期間－						
				入札・調達案件提供	インターネット	⇔	インターネット	－－－閲覧－					低		－－－					
				指名業者通知	インターネット	→	インターネット	通知書受取	指名通知の受取		●－－－－ ●－－－－ ○国民－基盤 ○企業－基盤	相手認証 鍵共有 守秘 署名	高		－－－－－－	－－	電子証明書	有	有	無
								通知書確認									－電子証明書－ＩＣカードに記録－	有	有	有
			入札・調達	入札参加希望申請受付	インターネット	←	インターネット	入札参加希望申請	入札参加申込		●－－－－ ●－－－－ ○国民－基盤 ○企業－基盤	相手認証 鍵共有 守秘 署名	高	－法令などにより定められる期間－	－－－	－署名－ －－－－－－－－－－ ハッシュ： －－－－－ －－	電子証明書 －ＩＣカードに記録－	有	有	有
				申請書検証	－インターネット→認証局－										－－－－独自－	－－	電子証明書	有	有	無
				入札参加希望審査	－－－						－－－	守秘	高							
				入札書－見積書－受付	インターネット	←	インターネット	入札書－見積書－入力	入札・見積金額入力		●－－－－ ●－－－－ ○国民－基盤 ○企業－基盤	相手認証 鍵共有 守秘 署名	高		－－－－－－－－－－ －鍵配信－ －－－－－－－－－－ －－	電子証明書 －ＩＣカードに記録－	有	有	有
				入札書－見積書－検証	－インターネット→認証局－										－－－－独自－	－－	電子証明書	有	有	無
				質疑応答－回答－	インターネット	⇔	インターネット	質疑応答－入力－			●－－－－ ●－－－－	相手認証	低		－－－	－署名－ －－－－－－－－－－ ハッシュ： －－－－－ －－	電子証明書	有	有	有
				入札案件進捗状況表示	インターネット	⇔	インターネット	－－－閲覧－					低		－－－					
				予定価格入力－登録－修正－ 開札執行 開札結果登録 抽選 再入札－再見積－	－－－						－－－	守秘	高							
				入札結果通知	インターネット	→	インターネット	通知書受取	結果受取		●－－－－ ●－－－－ ○国民－基盤 ○企業－基盤	相手認証 鍵共有 守秘 署名	高		－－－－－－	－－	電子証明書	有	有	無
								通知書確認									－電子証明書－ＩＣカードに記録－	有	有	有
			入札・調達結果の提供	入札結果登録	－－－				入札・調達状況閲覧		－－－	守秘	高	－法令などにより定められる期間－						
				入札結果公開	インターネット	⇔	インターネット	－－－閲覧－					低		－－－					
			契約						契約											
			支払						受領											

－－ 通信は、「－－－－－－－－－－鍵配信－－－－－－－－－－」で暗号化。

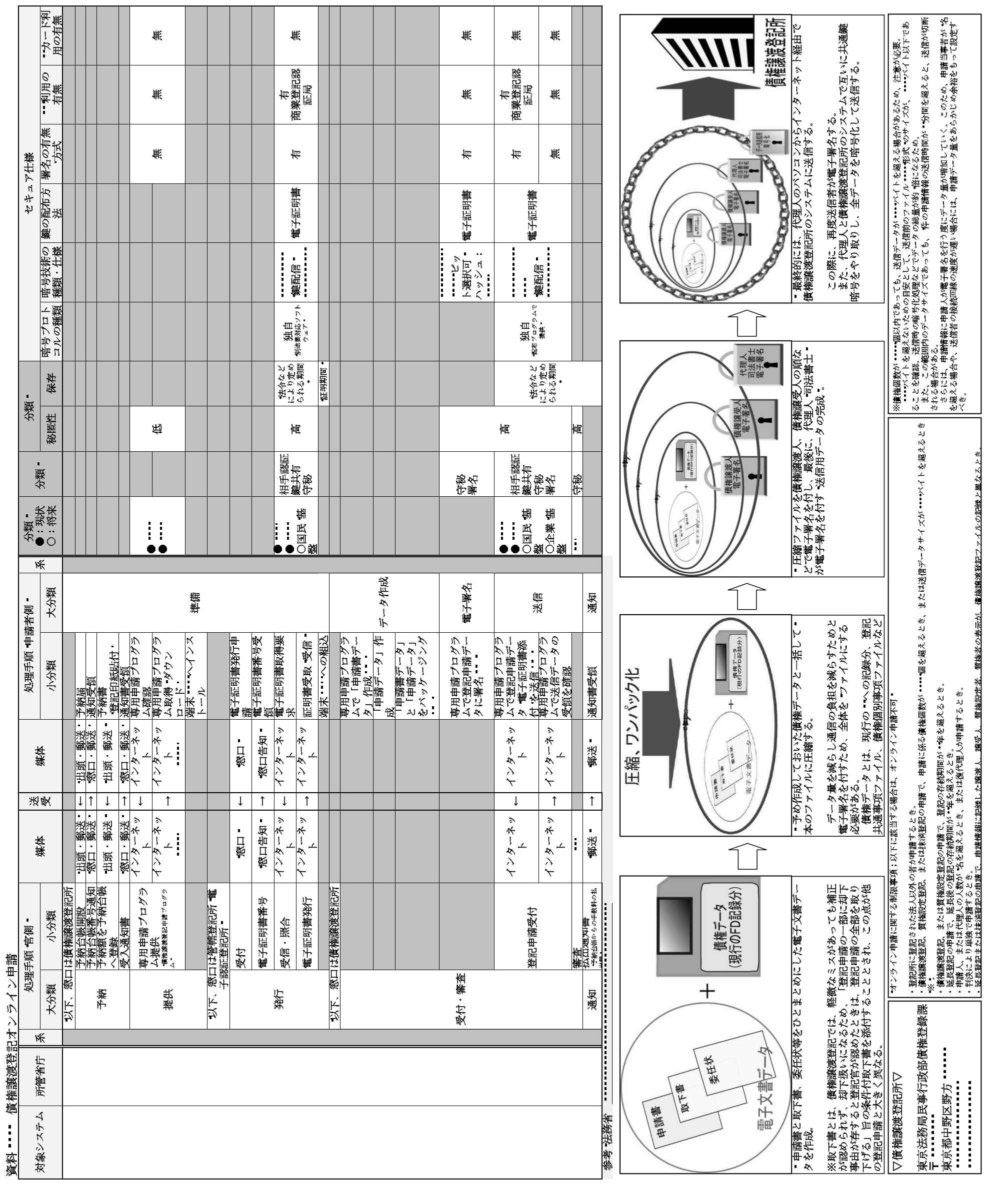


表 8: 債権譲渡登記オンライン申請システム

上に述べた各種電子政府システムで用いられる暗号プロトコルについてまとめたものを表 9 に示す。。

電子政府システム	暗号プロトコル関連	備考
電子申請システム	独自プロトコル/SSL	独自プロトコルは、事前プログラム配信
電子調達システム (コアシステム)	SSL, S/MIME	
電子調達システム共同運営システム（兵庫県）	SSL, S/MIME	
債権譲渡登記オンライン申請システム	独自プロトコル	独自プロトコルは事前プログラム配信
国税電子申請・納税システム	XML セキュリティ	XML 署名を利用
住民基本台帳カード	ISO14443-4	基盤技術：非接触ＩＣカード伝送プロトコル
政府認証基盤（GPKI）	ＰＫＩ	基盤技術：運用管理プロトコルについての利用は未規定
マルチペイメントネットワーク	IPsec	

表 9: 各電子政府システムにおける暗号プロトコル利用

3 関連技術調査

3.1 IPsec

PGP や S/MIME が電子メールのセキュリティ、SSH がリモート・ログインやリモート・コマンドのセキュリティ、SSL が Socket インタフェースを利用するアプリケーションのトランスポート層でのセキュリティをそれぞれ個別に実現するのに対し、IPsec は IPv4 および IPv6 による通信に汎用的に利用できるネットワーク層のセキュリティ・プロトコルとして IETF で標準化された。IPsec はまず 1995 年に RFC1825〜1829 として RFC 化され、改訂版が 1998 年に RFC2401〜2412 および 2451 として RFC 化されている (表 10)。また、現在も IETF の IPsec WG において AH/ESP 仕様の改訂、AES などの新しい暗号アルゴリズムの使用方法、および、新しい鍵交換プロトコルなどについて検討が続けられている。

IPsec では、IPv4 および IPv6 のデータグラム単位での認証、完全性の検査、リプレイ攻撃に対する防御、通信内容の秘匿などを実現する、AH(Authentication Header) と ESP(Encapsulating SecurityPayload) の２つのプロトコルと、IKE (InternetKey Exchange) プロトコル等の鍵管理の仕組みを組み合わせることでセキュリティを実現している。暗号化やメッセージ認証に用いるアルゴリズムに関する仕様は AH や ESP の仕様からは分離されており、新しいアル

ゴリズムに対して、AH や ESP での使用方法が仕様として定まれば、AH や ESP のプロトコル自体を改変することなく新たな暗号アルゴリズムが利用可能となる。なお、相互運用性の確保のため、各実装においてサポートが必須となるアルゴリズムが AH、ESP 等の各プロトコル毎に定められている。

IPsec で利用するプロトコル (AH,ESP) とそのオプション、暗号アルゴリズム、鍵等のパラメータは、IPsec を用いた通信の開始に先立ち、通信を行う双方のノードに事前に設定しておくか、IKE 等の鍵交換プロトコルを用いた交渉によって決められ、これを Security Association(SA) として各ノードが自身の持つ Security Association Database(SAD) 上に保持する。SA はアドレスや上位層プロトコルの情報に基づき、粒度の異なる様々な通信フローに対して個別に設定可能であり、どの通信に対し、どういった条件での SA の確立を必要とするかは、各 IPsec ノードの保持する Security Policy Database(SPD) 上のルールによって規定される。IPsec のポリシーや SA は送信方向と受信方向で個別に設定されるため、送信と受信で異なるプロトコルやアルゴリズムを利用することも可能である。以下では、IPsec を構成する AH, ESP, IKE の各プロトコルについて解説した後、IPsec の現在の仕様における問題点について解説する。

RFC 番号	タイトル
2401	Security Architecture for the Internet Protocol
2402	IP Authentication Header
2403	The Use of HMAC-MD5-96 within ESP and AH
2404	The Use of HMAC-SHA-1-96 within ESP and AH
2405	The ESP DES-CBC Cipher Algorithm With Explicit IV
2406	IP Encapsulating Security Payload (ESP)
2407	The Internet IP Security Domain of Interpretation for ISAKMP
2408	Internet Security Association and Key Management Protocol (ISAKMP)
2409	The Internet Key Exchange (IKE)
2410	The NULL Encryption Algorithm and Its Use With IPsec
2411	IP Security Document Roadmap
2412	The OAKLEY Key Determination Protocol
2451	The ESP CBC-Mode Cipher Algorithms

表 10: 1998 年に発行された IPsec 関連の RFC

3.1.1 Authentication Header (AH)

AH は IP データグラム単位での完全性の保証や送信元ノードの認証サービスに加え、オプションとして anti-replay サービスを提供するが、暗号化は行わない。AH のフォーマットは図 2 の通りであり、この AH が元の IP データ

グラムの IP ヘッダの後に挿入されるトランスポート・モードと、IP ヘッダを含めた元の IP データグラム全体の前に、新たに IP ヘッダと AH が付加されるトンネル・モードの 2 種類の動作モードがある。

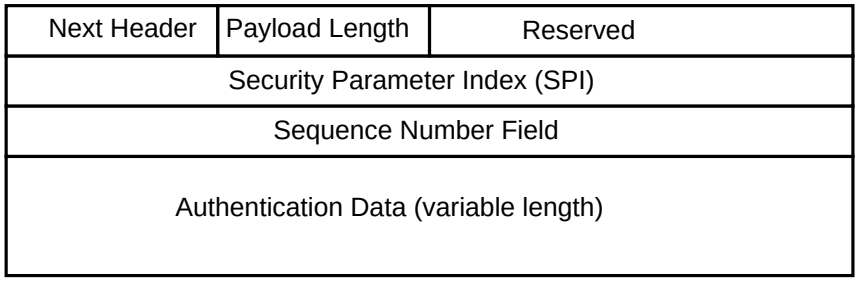


図 2: AH のフォーマット

Next Header フィールドは AH のあとに続くペイロード部分を識別するプロトコル番号で、Payload Length は 32bit ワード単位で表した AH の長さから (IPv6 の拡張ヘッダの規約に従い)2 を減じた値となる。SPI は IP ヘッダの送信元/ 送信先アドレスと組み合わせ、そのパケットの属する SA を識別するために用いられる 32bit の値である。Sequence Number は anti-replay 用のカウンタで、anti-replay サービスを利用しない場合、受信側ではこのフィールドは無視される。

Authentication Data には、IP ヘッダとペイロード部分に対し、SA で指定されたアルゴリズムによって計算された IP データグラムの Integrity Check Value(ICV) が格納される。AH の仕様では少なくとも HMAC-MD5 と HMAC-SHA-1 を実装する必要があると定められている。IP ヘッダ中の以下のフィールドは、パケット転送の際に変化するため、ICV の算出の際には全て 0 とみなして計算される。

- IPv4 の場合
  - TTL
  - Flags
  - Fragment offset
  - TTL
  - Header checksum
  - Options
- IPv6 の場合
  - Priority
  - Flow label
  - Hop limit

3.1.2 Encapsulating Security Payload

ESP は IP データグラム単位での完全性の保証や送信元ノードの認証、データの秘匿に加え、オプションとして anti-replay サービスを提供する。ESP のフォーマットは図 3 の通りであり、元の IP データグラムのペイロード部分を対象として暗号化や認証を行うトランスポート・モードと、元の IP データグラム全体を対象に暗号化や認証を施した ESP を作成し、新たに作成した IP データグラムのペイロードとして送信するトンネル・モードの 2 種類の動作モードがある。

SPI、Sequence Number は AH と同様で、Payload Data には、トランスポート・モードの場合、元の IP データグラムのペイロード部分が、トンネル・モードの場合、元の IP データグラム全体が SPI で指定されたアルゴリズムによって暗号化されて格納される。IV を使用する暗号アルゴリズムが用いられる場合には、ペイロード部分の各アルゴリズムの使用法に関する仕様の中で規定される位置に IV が格納されて送信される。Authentication Data はオプションで、ESP での認証サービスを利用する際に使われる。AH の認証サービスが先頭の IP ヘッダも対象に含めるのに対して、ESP の認証サービスは先頭の IP ヘッダは対象に含まれない。

ESP の仕様では認証用のアルゴリズムとして HMAC-MD5 と HMAC-SHA-1 が、暗号用のアルゴリズムとして DES-CBC の実装が義務づけられている。

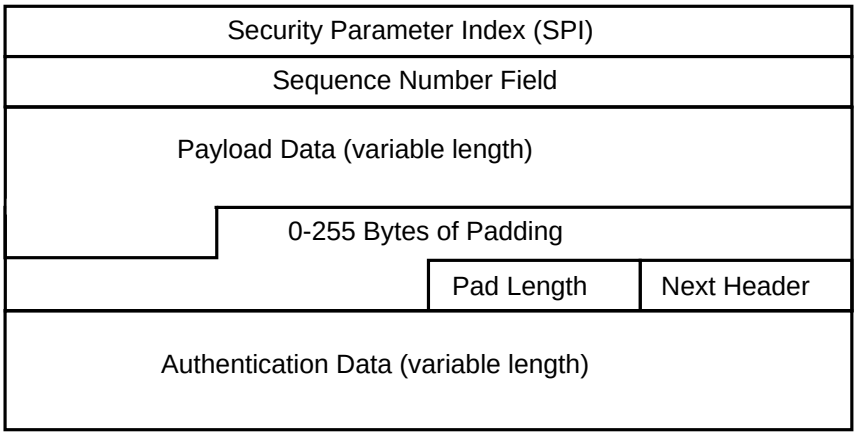


図 3: ESP のフォーマット

3.1.3 Internet Key Exchange (IKE) Protocol

IPsec のアーキテクチャは、任意の鍵交換プロトコルを利用して SA を確立することが可能な枠組みとなっているが、現在のところ IPsec で利用可能なものとして標準化されている鍵交換プロトコルは IKE のみである。

IKE は 2 つのフェーズからなり、フェーズ 1 で相互認証と IKE の通信を暗号化するための鍵の交換を行い (IKE 用の SA の確立)、フェーズ 2 において、IPsec で用いる鍵の交換を行う構成になっている。通信中に IPsec のセッション鍵の更新を行う場合には、フェーズ 2 のみが再度実行されるため、IPsec で用いる鍵を頻繁に更新する場合や、同一のノード間でアプリケーション毎に複数の IPsec SA を確立する場合などに効率が良いとされる。

フェーズ 1 には、3 つのメッセージで処理を完了するアグレッシブ・モード (図 4) と、6 つのメッセージを用いて、使用する暗号アルゴリズムに関する交渉や、各ノードのアイデンティティの秘匿も実現するメイン・モード (図 5) がある。また、認証に用いる事前共有鍵、公開鍵による署名、公開鍵による暗号 (新・旧[1]) の 4 種類の方式それぞれに対してプロトコルが規定されているため、計 8 通りのプロトコルが存在する。但し、このうち実装が必須とされているのは事前共有鍵を用いたメインモードのみである。

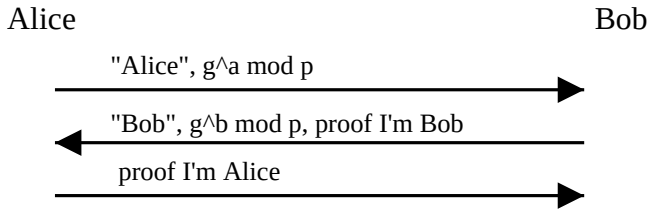


図 4: IKE のアグレッシブ・モード

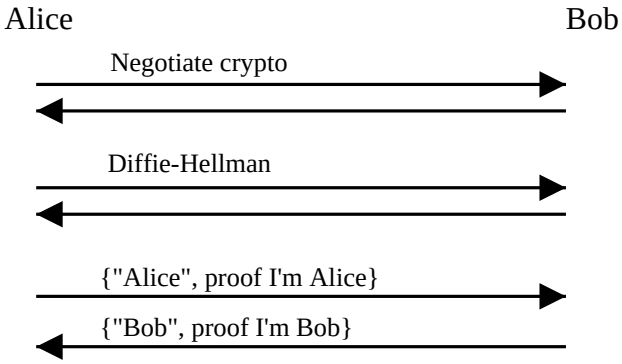


図 5: IKE のメイン・モード

フェーズ 2 ではクイック・モードと呼ばれるモードが使われ、フェーズ 1 で交換された共通鍵を用いて IPsec のセッション鍵の交換を行う。クイック・モードではオプションとして、Diffie Hellman 鍵交換を再度行い、perfect forward secrecy を実現することも可能である。ただし、この場合にはフェーズ 2 の処

[1]仕様の策定過程で方式が変更されたが互換 性維持のため古い方式も仕様に残されている。

理がさほど軽量とはならないため、セッション鍵更新のオーバヘッドを軽減するためにフェーズを 2 つに分割したことの意義が薄れると言える。

他に、フェーズ 1,2 には分類されないが、フェーズ 1 に引き続いて行われ、Diffie Hellman 鍵交換のための Diffie Hellman 群を決定するためのニュー・グループ・モードが用意されている。

IKE は Internet Security Association and Key Management Protocol (ISAKMP, RFC2408) によって定められているメッセージフォーマットを用いてメッセージ交換を行う。以下に、表 11 に示す略号を用いて、各モードの概要を説明する。

HDR	ISAKMP ヘッダ。ペイロードが暗号化される場合には HDR*と表記。
SA	1 つ以上のプロポーザルを含む SA ネゴシエーション・ペイロード。
KE	鍵交換ペイロード。
IDx	識別ペイロード。x は ii(ISAKMP initiator) または ir (ISAKMP responder)
HASH	ハッシュ・ペイロード。
SIG	署名ペイロード。
AUTH	SIG や HASH などの認証機構一般。
CERT	証明書ペイロード。
Nx	nonce ペイロード。x は i (initiator) または r (responder)
<P>_b	ペイロード<P>のボディ。
CKY-I と CKY-R	ISAKMP ヘッダ中にある initiator と responder のクッキー。
$g_i{}^x$	initiator の公開 Diffie-Hellman value
$g_r{}^x$	responder の公開 Diffie-Hellman value
prf(key,msg)	鍵 key とメッセージ msg を入力とする疑似乱数生成関数。
SKEYSTR	ピア同士のみが知っている秘密の鍵情報。
SKEYSTR_a	ISAKMP メッセージの認証に用いられる鍵情報。
SKEYSTR_d	鍵生成に用いられる鍵情報。
→	initiator から responder への通信を表す。
←	responder から initiator への通信を表す。
x|y	x と y の連結を表す。
[x]	x がオプションであることを表す。

表 11: IKE のメッセージ・フォーマット中の略号

メイン・モード 図 6 はメイン・モードにおいて交換されるメッセージの例である。(1),(2) において SA のプロポーザルとそれに対する応答、(3),(4) において Diffie Hellman の鍵確立、(5),(6) において相互認証が行われる。(5),(6) のメッセージは (3),(4) で交換された情報を基に生成された鍵により暗号化されるため、盗聴者に対してピアのアイデンティティは秘匿される。

アグレッシブ・モード 図 7 はアグレッシブ・モードにおいて交換されるメッセージの例である。メイン・モードと異なり 3 つのメッセージで完了す

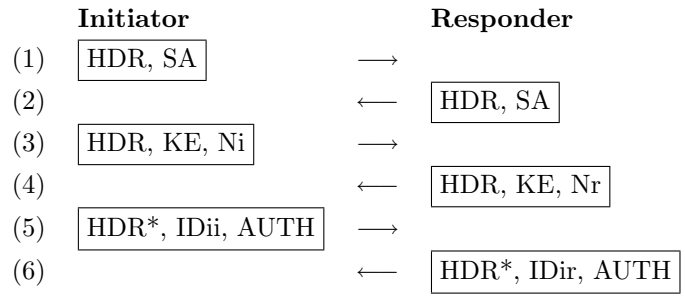


図 6: メイン・モードにおけるメッセージ交換の例

るが、ピアのアイデンティティは盗聴者に対して秘匿されない。

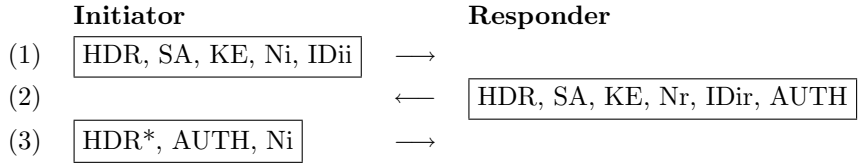
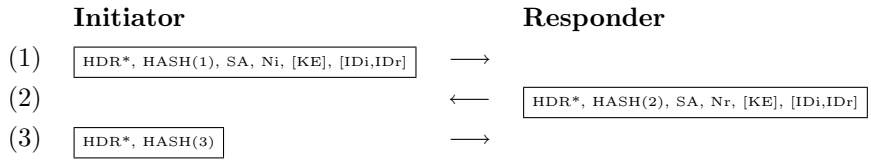


図 7: アグレッシブ・モードにおけるメッセージ交換の例

クイック・モード 図 8 はクイック・モードにおいて交換されるメッセージの例である。perfect forward secrecy が必要な場合には (1),(2) に KE ペイロードが含められ、新しい鍵情報 (NewKEYSTR) が以下により計算される。

$$\text{NewKEYSTR} = \text{prf}(\text{SKEYSTR\_d}, g^{xy}|\text{protocol}|\text{SPI}|\text{NI\_b}|\text{Ni\_r})$$



$$\text{HASH(1)} = \text{prf}(\text{SKEYSTR\_a}, \text{MsgID}|\text{SA}|\text{Ni}[|\text{KE}][|\text{IDi}|\text{IDr}])$$
$$\text{HASH(2)} = \text{prf}(\text{SKEYSTR\_a}, \text{MsgID}|\text{Ni\_b}|\text{SA}|\text{Nr}[|\text{KE}][|\text{IDi}|\text{IDr}])$$
$$\text{HASH(3)} = \text{prf}(\text{SKEYSTR\_a}, 0|\text{MsgID}|\text{Ni\_b}|\text{Nr\_b})$$

図 8: クイック・モードにおけるメッセージ交換の例

なお、現在の IKE は仕様があまりにも複雑であり、効率や安全性の面での問題を指摘されているため、IETF ipsec WG では現在 IKE をよりシンプルにした IKEv2 の仕様の策定を進めている。

3.1.4 IPsec の問題

IPsec の問題としては、第一にその仕様の複雑さが挙げられている [1]。AH、ESP、IKE 等の仕様では、複数の動作モードや多数のオプションが定められており、その組合わせの数の多さがプロトコルの理解や検証を困難なものとしているほか、相互運用性など実装上の問題や、設定誤りによるセキュリティ・レベルの低下などの運用上の問題を引き起こす要因となることが指摘されている。また IKE における DoS 攻撃対策の不完全さなど、仕様自体にも問題は存在している。

以下では、IPsec の仕様の持つ問題とそれに起因する運用上の問題、また、現在までに報告されている実装上の問題について述べる。

完全性チェックを伴わない ESP の使用

完全性チェックを行わない CBC モード暗号に対しては、以下に述べる Cut & Paste 攻撃 [2] によるメッセージの解読や改竄が可能であるため、ESP の利用に際しては暗号化だけでなく完全性チェックも行うことが望ましいとされる。

ところが、IPsec では、アプリケーション個別に SA を設定することが可能なことから、上位層で完全性チェックを行っている場合の冗長性を排除できるよう、ESP における完全性チェックをオプション扱いとしている。RFC2405 では、こうした形で ESP を利用しないことを強く推奨しているが、仕様上は AH および ESP での完全性チェックを伴わない ESP の使用が容認されているため、運用上の注意が必要である。

なお、現在ドラフト段階にある ESP 仕様の改訂版においては、現在その実装が必須 (「MUST」) となっている暗号化のみのサービスが、実装しても良い (「MAY」) という扱いに改められている。

- Cut & Paste 攻撃

CBC モードの暗号では、エラーの波及は 1 ブロックのみであるため、同一の鍵で暗号化された 2 つのメッセージを継ぎ合わせると、継ぎ目の 1 ブロックは正しく復号されないが、残りのメッセージは正しく復号される。このことから、IPsec で通信するホストの OS がマルチ・ユーザ OS であり、攻撃者が双方のホストにアカウントを持っている場合や、IPsec VPN で接続されたネットワークにおいて、双方のネットワークに接続可能である場合、攻撃者は盗聴や改竄の対象となる通信と同一の鍵で暗号化されたメッセージを送受信することができ、以下の攻撃が実行可能となる。

盗聴の場合には、攻撃者が盗聴対象のパケットの ESP ペイロード部分をコピーし、それとは別に攻撃者が攻撃者自身に宛てて送信し暗号化された UDP パケットの ESP ペイロード部分に、必要に応じてパケット長を合わせるため

のパディングを付加して貼り付けることにより、コピーした ESP ペイロードの内容を、最初の 1 ブロックを除いて正しく復号させることが出来る。

同様にして改竄の場合、攻撃者が送信して暗号化されたパケットの ESP ペイロードを、改竄対象となるパケットに必要なパディングと共に貼り付けると、最初の 1 ブロックを除いた改竄部分が正しく復号され正規の受信者に届く。例えば、telnet のセッション・ハイジャックの場合、正しく復号されない先頭ブロックの後に続いて、シェル・プロンプトに復帰するためのキー入力のシーケンスを書き込み、続いて実行させたいコマンドを書き込むことで、攻撃対象ホスト上で、任意のコマンドを実行させることも可能となる。

この攻撃は、AH もしくは ESP で完全性チェックを行うことで容易に排除可能である。

ESP における予測可能な IV の使用

ESP における DES の使用法を規定している RFC2405 には、IV の選定方法に関する要求条件の記述があり、そこでは IV のランダム性は要求されているが、予測不可能性は必要とされておらず、送信済の暗号ブロックの最終ブロックを次のパケットの暗号化の際の IV として利用することが仕様の中で容認されている。

ESP の暗号アルゴリズムに CBC を用い、上記のように IV が予測可能であるような実装がなされている場合、Cut & Paste 攻撃の場合と同様、攻撃者が、盗聴の対象となる通信と同一の鍵で暗号化されたメッセージを送信することができるならば、以下の方法で適応的選択平文攻撃による暗号解読が可能となる [3]。

ブロック暗号関数を F、その鍵を K、解読対象となる平文の各ブロックを $p_1, p_2, \ldots, p_n$ とすると、暗号化された各ブロック $c_1, c_2, \ldots, c_n$ は $c_0 = iv$ として、

$$c_i = F(K, p_1 \oplus c_{i-1}), \text{ただし } 1 \leq i \leq n$$

となる。このとき、次に送信される IP データグラムを暗号化する際の iv' が予測可能 (例えば $iv' = c_n$ とする実装の場合) であれば[2]、攻撃者は解読対象となる暗号ブロック c_j の平文 $p_j (c_j = F(K, p_j \oplus c_{j-1}))$ に対する自分の推測 G が正しいか否かを、以下の平文ブロック p' を作成して送信することにより確認出来る。

$$p' = G \oplus c_{j-1} \oplus iv'$$

この平文ブロック p' が暗号化された結果得られる暗号ブロック c' は

$$c' = F(K, G \oplus c_{j-1} \oplus iv' \oplus iv') = F(K, G \oplus c_{j-1})$$

[2]ESP で DES を使用した場合には、iv は平文で送信されるため、攻撃者は予測の正しさを検証できる

であり、$c' = c_j$ であれば $G = p_j$ となるため、推測の正否を判定出来る。解読対象が、パスワードを 1 文字送信する TCP パケットで、TCP ヘッダの予測が可能である場合など、解読対象のブロックのエントロピーが極めて低い場合に Brute force 攻撃による解読が可能となる他、メッセージに特定の文字列が含まれているか否かの判定なども可能となる。

p' が IP データグラムもしくはトランスポートヘッダとしての必要条件 (IP バージョン・フィールドの値は 4 か 6 でなければならない等) を満たしておらず、IPsec ノードの IP モジュールで受けつけられない場合、攻撃者は条件を満たすような他の予測値を試すか、条件を満たす p' を作成できるような IV になるまで、通常のパケットを送信して IV を更新すれば良い。

この攻撃法に対する脆弱性の指摘により、現在ドラフト段階にある “The AES Cipher Algorithms and Their Use With IPsec” においては、IV の選定の際に必要とされる要件として、予測不可能性が明記されたが、現在のところ DES の IPsec における使用法に関する仕様の改訂は行われていない。

リプレイ攻撃

IPsec では、オプションとして 32 ビットのシーケンス番号を用いた anti-replay サービスを選択可能である。ただし、IKE 等の鍵交換プロトコルを使わずに、手動で鍵を設定する場合には、32bit のシーケンス番号が一巡する前に自動的に鍵を更新することが出来ないことから、anti-replay サービスは利用できないと仕様で定められており[3]、リプレイ攻撃に対して脆弱性を持つ。

弱い暗号アルゴリズムの使用

ESP の仕様で実装を必須としているペイロード暗号化のためのアルゴリズムは、今日ではその強度に疑問のある DES のみであり、他のアルゴリズムをサポートするか否かはベンダの選択に委ねられている。そのため運用上の問題として、通信相手も含めた、IPsec 機器やソフトウェアにおける種々のアルゴリズムのサポート状況とその設定に関して十分に注意を払わなければ、期待するレベルのセキュリティを得られない可能性がある。

IKE に対する DoS 攻撃

IKE の Phase 1 は大量の計算資源を必要とするため、DoS 攻撃への対処が必須である。このため、IKE では、最初の接続要求に対して Cookie を送り、同一の正しい Cookie を送り返してきた相手に対し、その後の処理を継続することで、偽の SA 確立要求に対して計算機資源を浪費しないようにしている。ところが IKE の Cookie 処理では、接続を受け付ける側が、接続要求を受けて Cookie を送り返す際にその状態を記憶しておく必要があり、メモリの

[3]不完全な形で anti-replay サービスを提供するよりは、そもそも提供しない方が良いとの判断。

大量消費を狙った DoS 攻撃に対して脆弱性を持っている [5]。現在策定中の IKEv2 では Cookie がステートレスなものとなり、こうした攻撃への脆弱性は排除されている。

IKE における identity の漏洩

IKE において、ノードの identity を第三者から秘匿したい場合にはメイン・モードが使われるが、メイン・モードにおいて、事前共有鍵を用いる場合には、ノードの identity 情報として使えるのが IP アドレスのみであるため、identity の秘匿は行われない。また、メイン・モードで公開鍵署名を用いる場合でも、図 9 の 4 番目のメッセージまでのやりとりでは相手の認証が出来ないため、5 番目のメッセージにおいて、IKE セッションを開始した Alice(通常クライアント側) の identity が、Bob になりすました攻撃者に対して漏洩する可能性がある [4]。なお、公開鍵暗号を用いる場合には identity の漏洩は生じない。

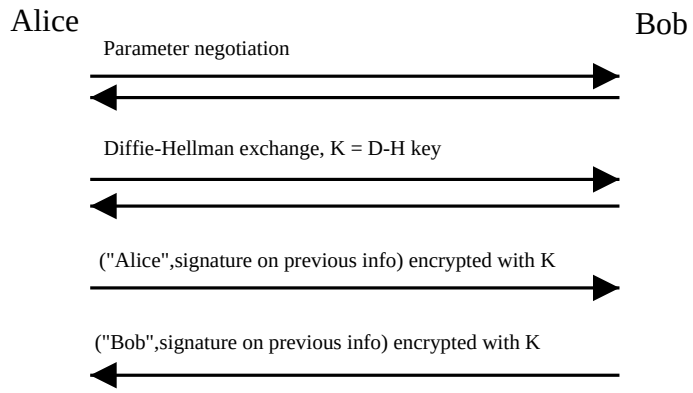


図 9: 公開鍵署名を用いたメイン・モードにおける identity の漏洩

アプリケーションからみた IPsec

IPsec は上位層のプロトコルとは独立に動作するため、OS やネットワーク機器において IPsec がサポートされていれば、既存のアプリケーションを変更しなくても安全な通信を行うことが可能となる。逆に、アプリケーションを変更しない場合、アプリケーションや利用者において安全に通信が行われているか否かを判別する術がない。そもそも、IPsec の設定や状態の確認をするための標準的な API が確立されていないため、現状ではアプリケーション自身が通信の安全性を検証するのは困難である。特に中継ノードがトンネル・モードを用いてネットワーク間を接続する IPsec VPN のような利用形態の場合、現在のところエンド・ノードから IPsec の動作を制御する手段が存在していない。

こうしたことから、IPsec のポリシー設定の誤り等により、IPsec による保護のない状態で通信が行われた場合でも、それをアプリケーションが検知出来ないという危険性があり、アプリケーション、OS、中継ノード等の運用形態や管理権限の所在によっては、アプリケーション個別のセキュリティ技術や、アプリケーションから直接制御可能な SSL 等の手段を利用するのが適切な場合もある。

このように、IPsec はアプリケーションやトランスポート層プロトコルによらず汎用的に使える技術であるが、IPsec の利用がセキュリティを実現するのに必ずしも最適な方法とは限らないことに注意が必要であり、IETF の ipsec WG においても、セキュリティの実現を IPsec に依拠するようなプロトコルに対するガイドラインの策定を進めている。

過去に報告された実装上の問題

以下に、IPsec に関して過去に報告された実装上の問題を列挙する。なお、いずれの問題もベンダが提供するパッチや最新版への更新によって解決可能となっている。

- Multiple vendors' IKE implementations do not properly handle IKE response packets (2002-09-17)

Cisco, NetScreen, Network Associates, OpenBSD, PGP, SafeNet 等の多くの IKE の実装において、例外的なレスポンス・パケットを正しく処理しないことにより以下に挙げるいくつかの脆弱性が報告された。

- レスポンス・パケットの SPI の値が過度に大きい場合にバッファ・オーバフローを引き起こす
- レスポンス・パケットのペイロード数が過度に多いか、ペイロードのサイズが過度に大きい場合にバッファ・オーバフローを引き起こす。
- ペイロード長が 0 のレスポンス・パケットを受け取った際に、CPU 資源を消費するため、これを狙った DoS 攻撃が可能となる。

- Multiple IPsec implementations do not adequately validate authentication data (2002-10-17)

KAME(FreeBSD, NetBSD)、 FreeS/WAN(Linux) 等の実装において、非常に小さいパケットの認証フィールドの長さが正しく計算されず、値を保持する unsigned interger のオーバーフローする。これによって、認証データの計算対象となるメモリ領域が非常に大きなものとなり、カーネル・パニックを引き起こす。

- NetBSD IPsec ESP vulnerability (2002-10-22)

NetBSD 1.5, 1.6 および current (2002-10-22 現在) において、IPsec でのパケット長の検査が十分に行われないため、特殊な ESP パケットによりカーネル・パニックを引き起こし、システム全体を停止させられる危険性がある。

3.2 蓄積メッセージへの署名と暗号化 (S/MIME, XML Security)

SSL や IPsec では通信が行われている際に通信相手の認証やメッセージの暗号化はできても、通信後にディスク等に蓄積された後のコンテンツに対してはセキュリティの保証が出来ない。このため、蓄積して転送されるメッセージに対しては上位層での署名や暗号化の仕組みが必要となり、アプリケーション毎に S/MIME(主に電子メール) や XML Signature, XML Encryption などの標準が規定されている。

SSL や IPsec などの場合と異なり、蓄積されるメッセージに対して署名と暗号化を同時に行う場合には、不正な転送に対する注意が必要となる [6, 7, 8]。このため、以下ではまず、S/MIME と XML Security に共通する、不正な転送の問題について述べた後、S/MIME と XML Security それぞれの概要と問題について述べる。

3.2.1 不正な転送に対する脆弱性

共通鍵を用いたメッセージの暗号化においては、暗号化を行える人物は共通鍵の所有者のみであるため、署名と暗号化が行われたメッセージについて、署名者と暗号化を行った人物の同一性は容易に確認できる。これに対して、公開鍵を用いてメッセージの暗号化を行った場合、公開鍵は誰もが入手可能であるため、受信者側では、誰によって暗号化されたメッセージであるかを判断できない。ところが一般的な利用者はこの違いを明確に意識せず、署名と公開鍵による暗号化が施されたメッセージを受信した際に、共通鍵の場合と同様、署名と暗号化が同一の人物によってなされたものと解釈しがちである。

このため、署名を施したメッセージに対して公開鍵を用いて暗号化を行う場合や、もしくは公開鍵で暗号化を行ったメッセージに対して署名を施す場合、署名と暗号化が相互に依存しないような単純な形で両者を組み合わせると、不正な転送に対する (利用者側での間違った解釈に基づく) 脆弱性を持つ場合がある。

メッセージ $\{m\}$ が A により署名されたものを $\{m\}^a$、B の公開鍵を用いて暗号化されたものを $\{m\}^B$ とし、以下のように、A から B 宛の署名後に暗号化を施したメッセージを受信した B が、これを復号化した後、C の公開鍵で暗号化したとする。

$$\begin{array}{lll} \mathrm{A} \to \mathrm{B} & : & \{\{I\ love\ you\}^a\}^B \\ \mathrm{B} \to \mathrm{C} & : & \{\{I\ love\ you\}^a\}^C \end{array}$$

この場合、$\{\{I\ love\ you\}^a\}^C$ を C が受け取ると、これは A 自身が署名と暗号化を施したメッセージであり、A と C 以外は内容を知り得ないと間違って解釈する恐れがある。

逆に暗号化を施した後に署名を行った場合に、以下のように A から B に宛てたメッセージを C が横取りし、署名部分のみを書き換えたとする。

$$\begin{array}{lll} \mathrm{A} \to \mathrm{B} & : & \{\{my\ idea\}^B\}^a\ (\text{C が横取り}) \\ \mathrm{C} \to \mathrm{B} & : & \{\{my\ idea\}^B\}^c \end{array}$$

この場合、B は $\{\{my\ idea\}^B\}^c$ を C が作成したメッセージと解釈する恐れがある。

一般に、署名と暗号化を独立の処理とした場合にこうした脆弱性が生じるが、これは一方で署名や暗号化を多重にネストする場合の実装を容易にすることなどもあり、S/MIME や XML Signature/Encryption 等の仕様の策定に際してはこの問題について議論されつつも、以下のような理由により特別な対処はなされなかった。

- 署名を行う際に受信者名もしくはメールヘッダ全体を署名対象の本文に含めるためのオプションがある
- メッセージ本文に受信者の名前が登場する場合などに、不正転送は文脈から検知できる

署名と暗号化がなされたメッセージにおける署名者と暗号化を行った人物の同一性に関して、全ての利用者が共通鍵を用いた場合と公開鍵を用いた場合での違いを認識し、誤った解釈をしないことが期待できるならば、そもそもこうした不正な転送が問題となることはない。ところが一般的には誤って解釈する利用者の存在する可能性が否定できないため、送信者の側としては、メッセージ本文や、オプションの属性を用いて署名対象に受信者名を含める (署名後に暗号化の場合)、暗号化するメッセージ本文に送信者名を入れる (暗号化後に署名の場合)、などの対処により、不正な転送が行われた場合にそれを受信者側で明確に検知出来るようにすることが望ましい。

3.2.2 S/MIME

S/MIME は Cryptographic Message Syntax(CMS, RFC2630, のちに RFC3369 と RFC3370 で改訂され、構文とアルゴリズムに関する仕様が分離) を利用して MIME メッセージにおける単一もしくは複数の MIME パート、またはメッセージ全体に対して認証、完全性、秘匿を行うもので、電子メールに限らず、MIME をサポートする HTTP などの他の転送方式においても利用可

能である。S/MIME はバージョン 3 が最新の標準で、その仕様は RFC2632, 2633,2634 等で規定されているが、現在、サポートを必須とするアルゴリズムを変更するなどしたバージョン 3.1 の仕様の策定が IETF smime WG で進められている。

S/MIME では、署名／暗号化の対象となる MIME パートもしくはメッセージ全体に対し、CMS に従い図 10、図 11 の形式で署名もしくは暗号化を施す。署名／暗号化した MIME パートに対しては、MIME タイプとして application/pkcs7-mime を指定し、種類に応じて enveloped-data、signed-data、certs-only 等の smime-type パラメータを付加する。

```
CMSVersion
DigestAlgorithmIdentifiers

EncapsulatedContentInfo
(メッセージ本文)


CertificateSet (Optional)

SignerInfos
  CMSVersion
  SignerIdentifier
  DigestAlgorithmIdentifier
  SignedAttributes (Optional)
  SignatureValue
  UnsignedAttributes (Optional)
```

図 10: CMS による署名フォーマット

S/MIME では署名と暗号化は自由にネストさせることが可能であり、Enhanced Security Services for S/MIME (RFC2634) ではこれを利用し、エンドユーザが署名/暗号化したメッセージに対して更にメーリングリストサーバが署名を施す、セキュアメーリングリストなどの拡張サービスついても述べられている。

バージョン 3 でサポートを必須としているアルゴリズムは、メッセージダイジェストが SHA-1、署名が DSA、鍵交換が Diffie-Hellman、暗号化が 3DES であり、現在策定中のバージョン 3.1 においては鍵交換用アルゴリズムとして Diffie-Hellman の代りに RSA が必須のアルゴリズムとなっている。

```
CMSVersion
OriginatorInfo (Optional)
  CertificateSet (Optional)
  CertificateRevocationList (Optional)
RecipientInfos

EncryptedContentInfo
(暗号化されたメッセージ本文)




UnprotectedAttributes (Optional)
```

図 11: CMS による暗号化のフォーマット

“Small-Subgroup” Attaks on the Diffie-Hellman Key Agreement Method for S/MIME

Diffie-Hellman において、公開鍵としてオーダーの小さい不正な鍵を使用された場合に秘密鍵の情報が漏洩する脆弱性があり [9]、S/MIME においてこの攻撃への対処が必要となる場合と、その対処方法について RFC2785 で解説されている。

この攻撃は、不正な公開鍵を基にして作成されたセッション鍵による復号の成否を攻撃者が知り得るか、もしくは、セッション鍵によって生成された MAC または暗号文を入手可能な場合に成り立つため、S/MIME の場合にはその危険性を以下の場合に分類して考えることができる。

受信者側 実装がメッセージの復号エラーを送信者に通知しなければ、実装上は問題ないと言える。ただし、口頭での復号結果の確認、返信の有無、などにより復号の成否が送信者に伝わる場合には対処が必要となる。

送信者側 使用する鍵ペアの種別 (ephemeral/statick) により異なる。

ephemeral-static Diffie-Hellman 送信者が一度限りの鍵ペアを使用する場合には、秘密鍵の漏洩は実際上問題とならないので対処は不要。

static-static Diffie-Hellman 対処が必要。

対処法は以下の通り

- クライアントが相手の公開鍵の検証を行う。
- CA が公開鍵の証明書を発行する際に、正しく公開鍵の検証を行う。
- 素数 p を $p-1=2*q*k$(k は大きな (q と同等かそれ以上の) 素数か大きな素数の積) となるように選択し、オーダーの小さな不正鍵を生成すること自体を困難にする。
- Compatible Cofactor Exponentiation [10] を用いてセッション鍵を生成する。
- Non-compatible Cofactor Exponentiation [10] を用いてセッション鍵を生成する。

Million Message Attack on CMS

CMS で RSA によりコンテンツ暗号鍵を暗号化する際、PKCS#1 v1.5 で規定されている方法を用いてパディングを行うため、SSL の場合と同様に Million Message Attack(MMA) に対する脆弱性をもつ。このため、RFC3218 においてその攻撃手法と対処法について解説されている。なお、この攻撃を成功させるためには数百万の復号を行う必要があるため、S/MIME の場合では、メーリングリストサーバのように、復号が自動化されている相手に対してのみ有効となる。

MMA は、PKCS#1 v1.5 のパディング規則において、メッセージブロックの先頭 2 バイトが'00 02' となり、その後に非 0 の乱数列、1 バイトの 0 が順に埋め込まれ、最後にメッセージ (コンテンツ暗号鍵) が埋め込まれることを利用している。その上で、MMA では攻撃対象に攻撃者が選択した暗号文を復号させた際に、復号結果の先頭 2 バイトが'00 02' となっているか否かを、復号処理におけるエラーメッセージにより判別できることを前提として選択暗号文攻撃を行う。具体的には、解読対象の暗号ブロック C を元に、一連の整数 S を用いて $C' = C*(S^e) mod\, n$ により作成した暗号ブロック C' を攻撃対象に復号させ、先頭 2 バイトが'00 02' となるような S を多数発見し (およそ 2^{16} 回の試行に 1 回見つかる)、これを用いて求める平文を得るという処理を行う。

SSL の場合の対処は、PKCS#1 v1.5 の代りに、Optimal Asymmetric Encryption Padding (OAEP) を利用する PKCS#1 v2.0 を用いることであったが、S/MIME の場合、これは既存の実装との相互運用性を損なうことになるので、RFC3218 では以下の 2 つの対処法が提案されている。

- 先頭 2 バイトをチェックすると同時に、コンテンツ暗号鍵の鍵長 (最初の 0 の位置から判別) やパリティビット (もしあれば) もチェックし、それらが不正であれば全て同一のエラーメッセージを返す。これにより、解読に利用出来る S をひとつ発見するのに 2^{16} よりも遥かに多くの試行が必要となり、攻撃の危険性が減少する。

- メッセージブロックのフォーマット、鍵長、パリティビット等が不正な場合には、コンテンツ暗号鍵に乱数をセットして以降の処理を続ける。これにより、攻撃者はRSA復号処理においてメッセージブロックのフォーマットが適正か否かを判別出来なくなり、攻撃が成立しなくなる。

過去に報告されたS/MIMEに関する実装上の問題

バッファオーバーフロー等の一般的なバグの他に、以下の不具合が報告されている。

- 証明書チェーンの詐称に対するMicrosoft Outlook S/MIMEの脆弱性
  Outlookにおいて証明書チェーンが正しく検証されないため、不正な証明書を有効なものとして受理してしまう。
- Lotus Notes R5 S/MIMEにおける改竄チェックの不具合
  署名つきのメッセージが改竄されていた場合に、Lotus Notesは改竄を検知したことをユーザに通知せずに、単なる署名なしメッセージとして表示する。また復号できない暗号メッセージを受け取った場合に、復号エラーをユーザに通知せずに、空のメッセージを表示する。

3.2.3 XML Security

S/MIMEがMIMEパートもしくはメール全体を単位とした署名や暗号化を行うのに対して、XML Securityでは XML文書の任意の要素や外部リソースなどを対象にしたより柔軟な署名／暗号化の仕組みが提供される。XML SignatureとXML Encryptionに関する基本仕様はいずれもW3Cの勧告として標準化されており、XML Signatureに関してはIETFのRFC2807,3075,3076,3275としても発行されている。

XML Signature

XML Signatureを用いた署名の構成要素は図12の通りで、以下の対象に対して署名を行うことができる。

- XML文書の全体または一部
- 署名の中に埋め込まれたXML要素
- URIで指すことのできる外部リソース
- 上記の任意の組み合わせ

XMLでは空白文字の有無、文字符号化方法、属性の順序などに自由度があるため、XML Signatureではハッシュ値を計算するのに先立ち、Canonical

XML(RFC3076) などで規定されている方法により署名対象となる XML 文書リソースを正規化し、同一コンテンツ間の表層的なバイト列の違いを吸収することとしている。

XML Signature において実装が必須とされているアルゴリズムは、ダイジェストに SHA1、MAC に HMAC-SHA1、および、署名に DSA-SHA1(RSA-SHA1 も実装を推奨) となっている。

```
<Signature>
  <SignedInfo>
    (CanonicalizationMethod) 署名対象の正規化に用いた方法
    (SignatureMethod)
    (<Reference (URI=)?> 署名対象を参照する URI
      (Transforms)? 署名を検証する前に参照対象に対して行う変換手順
      (DigestMethod)
      (DigestValue) 参照された文書リソースのダイジェスト
     </Reference>)+
  </SignedInfo>
  (SignatureValue) 署名の値
  (KeyInfo)? 署名の検証に用いる鍵の情報 (証明書、鍵の名前、鍵確立のアル
             ゴリズムとその情報など)
  (Object)*
</Signature>

?: 0 回または 1 回の出現
+: 1 回以上の出現
*: 0 回以上の出現
```

図 12: XML Signature の構成要素

XML Encryption

XML Encryption の XML 要素の構成は図 13 の通りで、下記の対象を EncryptedData として暗号化できる。

- XML 要素全体
- タグと属性を除く XML 要素の内容
- 任意の外部リソース
- 上記の任意の組み合わせ

XML Encryption において実装が必須とされているアルゴリズムは、ブロック暗号に 3DES と AES-128 および AES-256、鍵転送に RSA-v1.5 と RSA-OAEP となっている。

署名と暗号化は独立に行うことが可能であり、署名の後に署名対象の一部を暗号化することや、既に暗号化されている部分に対して署名を行うことなどが可能であり。このため、正しく署名の検証を行うためには、署名の検証の前に復号すべき部分と復号すべきでない部分を指定する必要があり、その方法 (Decrypt Transform) が W3C の勧告として出されている。Decrypt Transform の勧告では、署名時に暗号化されていた EncryptedData を指定し、署名検証時には指定のない EncryptedData を全て復号化することとしている。

```
<EncryptedData Id? Type? MimeType? Encoding?>
  <EncryptionMethod/>?
  <ds:KeyInfo> 鍵に関する情報
    <EncryptedKey>?
    <AgreementMethod>?
    <ds:KeyName>?
    <ds:RetrievalMethod>?
    <ds:*>?
  </ds:KeyInfo>?
  <CipherData>
    <CipherValue>? 暗号化されたデータ (直接埋め込む場合)
    <CipherReference URI?>? 暗号化されたデータへの URI(外部に置く場合)
  </CipherData>
  <EncryptionProperties>?
</EncryptedData>

?: 0 回または 1 回の出現
+: 1 回以上の出現
*: 0 回以上の出現
```

図 13: XML Encryption

XML Signature、XML Encryption ともに、仕様が定まってからの日が浅く、普及がこれからということもあり、現在のところ実装上の脆弱性が問題として一般に報告された例はみあたらない。

3.3 インターネットにおける PKI 関連プロトコル

X.509 は証明書のフォーマットやフィールド、公開鍵の配布手続きなどについて、その利用分野を問わない形での極めて汎用的な標準を定めているため、証明書の中身や PKI の動作モデルについて多くのバリエーションを許容している。このため、相互運用性の向上や、実装に要する負担の軽減を図るため、場合によっては利用分野毎に最低限必要とされる X.509 のサブセットを定義することが必要と考えられる。こうした観点から、IETF の pkix WG では、X.509 に基づく PKI をインターネットにおいて普及させるのに必要な標準の策定を行っている。

PKIX では PKI の構成要素とその関係を図 14 のようなものとして規定し、証明書フォーマットや、構成要素間の通信に用いられる操作プロトコル、管理プロトコル、証明書検証プロトコル等の各種プロトコルを定めている。

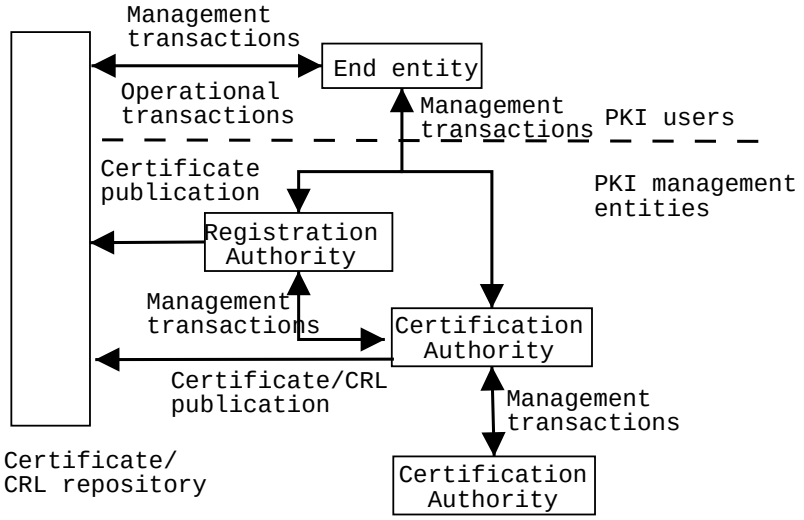


図 14: PKI の構成要素

3.3.1 Operational Protocols (LDAP, FTP, HTTP)

PKIX では、PKI の構成要素間で証明書や CRL 等の情報を交換する際に用いる下位の転送プロトコル毎に、そのプロトコルに対する要求仕様をまとめており、現在、LDAP に対して RFC2559、FTP と HTTP に対して RFC2585 が発行されている。

FTP や HTTP はサーバ、クライアント共に一般に普及しているため、証明書を、その URI を示して配布する場合には極めて有効であるが、URI が不明の証明書をメールアドレスなどを元に検索する手段がないため、LDAP を利用する方が利便性は高い。以下では、LDAP プロコトルについて、その概要と、PKIX での利用法について述べる。

LDAP

LDAP(Lightweight Directory Access Protocol) は、元々は TCP/IP ネットワークからゲートウェイを介して X.500 ディレクトリ・サービスを利用するためのプロトコルとして開発されたものであるが、現在は LDAP で直接ディレクトリ・サービスを提供する LDAP サーバを用いて、X.500 なしで利用されるのが一般的になっている。現在 LDAPv3 が RFC 2251-2256 および 2829-2831 によって規定されており、IETF の ldapbis WG において LDAPv3 の改訂作業が行われている。

LDAP サーバは X.500 同様図 15 のような DIT(Directory Information Tree) と称されるツリー上のデータ構造を用いてデータ・エントリを管理する。DIT 上のエントリは “cn=Suzuki Ichiro, ou=Engineering, o=abc, c=JP” のような DN(Distinguish Name) によって識別される。

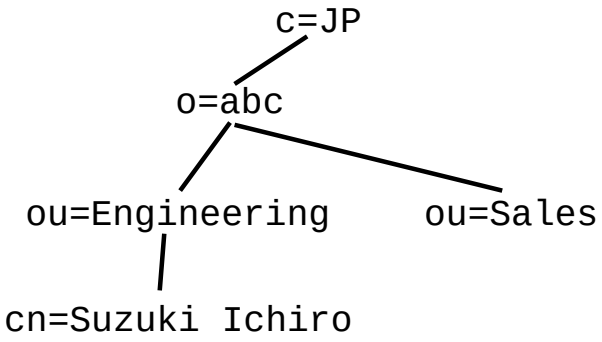


図 15: ディレクトリ情報ツリー (DIT)

DIT 上のエントリは属性 (型とその値の組) の集合を持ち、その中には少なくともひとつの objectClass 型の属性が含まれている。objectClass は、そのエントリが保持すべき、もしくは保持してよい属性の種類を規定している。objectClass が保持すべき属性の集合や、各属性に対して属性値として許される値、シンタックス、値の比較方法 (大文字/小文字を区別しない文字列比較、など) は schema によって定義される。

上記のデータを保持する LDAP サーバに対して、LDAP では以下の 6 つのオペレーションを定義している。

- サーバへのバインド
- サーチ
- エントリの比較
- エントリの追加
- 既存エントリの変更
- エントリの削除

LDAP では、DIT 上のエントリ毎に固有のアクセス権限を設定可能であり、バインド操作は、LDAP クライアントが、DIT 上の特定のエントリ (一般的には person オブジェクト等の人物エントリ) として LDAP サーバに接続するための操作である。バインドの際に特定のエントリを指定しない場合には anonymous 接続となり、読み出しのみ可能、などその後の操作に大きく制限を課せられのが一般的である。

サーチは、DIT の特定の場所を起点とし、属性値に関する条件式に合致するエントリを検索し、その DN と属性値の集合を返す。比較操作は、条件に合致するエントリの有無のみを調べる。

エントリの追加、変更、削除に関しては、その権限を細かく規定するのが一般的であり、例えば、一般ユーザのエントリとしてバインドした場合には、自分自身のパスワードなど特定の属性の変更のみが可能で、他の属性の追加・変更・削除、エントリの追加や削除は管理者としてバインドした場合のみ可能となるような設定がなされる。

LDAP のセキュリティに関しては、バインド操作によるアクセス制限、バインド操作の際の SASL(Simple Authentication and SecurityLayer) を用いた種々の認証方式のサポート、サーバ・クライアント間の通信への SSL/TLS の利用 (ldaps) 等の手段が提供されている。

PKIX における LDAP の利用に関しては、RFC2559 で LDAPv2 を PKI で利用する際に実装が必須となる LDAPv2 仕様のサブセットが定義されており、RFC2587 で PKI 用の LDAPv2 schema が定義されている他、現在 IETF pkix WG において LDAPv3 に関する schema の定義などについて検討が進められている。

PKIX における証明書配布の際の下位転送プロトコルとしての LDAP 利用に関しては、PKI のエンティティ間で送受されるデータが、それ自体に署名が施されており、公開情報でもあることから、通信内容の秘匿やメッセージの完全性チェックなどは基本的に不要としており、CA がリポジトリ上の証明書/CRL の情報を更新する際のバインド操作における認証を強力なものにすることを推奨している他には、セキュリティに関する要求はない。

3.3.2 Management Protocols (CMP, CMC)

PKIX では、管理メッセージのフォーマットを Certificate Request Message Format (CRMF, RFC2511) によって定め、管理メッセージの送受に関するプロトコルに関しては、Certificate Management Protocols (CMP, RFC2510) と Certificate Management Messages Over CMS (CMC, RFC2797) の 2 種のプロトコルを定めている。CMP と CMC はほぼ同等の機能を提供しているが、現在のところ両者が一本化される動きはなく、2 つの標準が並立する状態になっている。

Certificate Management Protocols (CMP)

CMP では、PKI の管理機能として以下を定義し、それらの機能を実現するに際して PKI のエンティティ間で送受されるメッセージについて、CRMF も利用しつつ新規にフォーマットを定めている。

- CA establishment

- End-entity initialization
- Certification
- Certificate and CRL discovery
- Recovery operations
- Revocation operations

CMP では、PKI のエンティティ間で送受されるメッセージに MAC もしくはデジタル署名を付加して完全性の保証やメッセージ発信者の認証を行うことが可能となっている。証明書の発行を初めて要求する利用者に対して、その利用者の認証を行う場合には、利用者と PKI(CA) の間で認証のための鍵 (Initial Authentication Key (IAK)) を CMP 以外 (out-of-band) の手段で共有し、IAK を用いて証明書発行要求のメッセージに MAC を付加することとなっている。

また、公開鍵の発行要求を処理する際には、要求者に対して、対応する秘密鍵を所有していることの証明 (Proof of Possession (POP)) を、CMP メッセージを用いて (in-band)、もしくは CMP 以外 (out-of-band) の手段で要求する必要があるが、CMP では in-band での方法を鍵の種類に応じて以下のように規定している。

Signature Keys 要求者が、証明書発行要求に対して秘密鍵を使って署名を施す

Encryption Keys 次の 3 つの方法のいずれか

- 秘密鍵そのものを暗号化して CA/RA に送信
- CA/RA が要求者に対して任意のメッセージの復号を要求する (challenge & response)
- CA/RA が証明書を要求者の公開鍵で暗号化して送信し、要求者はそれを復号できたことを示す適正な確認メッセージを返す

Key Agreement Keys Diffie-Hellman 鍵のような鍵共有のための鍵の場合、実際に要求者と CA/RA の間で鍵の共有を行う

一般に、任意の暗号文に対して復号結果を返すような振る舞いは、選択暗号文攻撃を可能とするため、セキュリティ上好ましくない。CMP の複号鍵の POP においては、challenge メッセージの認証を偽られるなど、複数の要因が重ならない限り攻撃は成立しないが、実装や運用に際してはこうした危険性に対して十分注意を払うべきであるとされている。

CMP メッセージの転送方式については、ファイル交換 (DER エンコードされたファイルを FTP 等で送受)、TCP の直接利用 (ポート番号 829)、電子

メールもしくは HTTP(pkixcmp MIME オブジェクトとしてで送受)、などの方法が規定されている。なお、CMP には先に述べたメッセージの完全性の保証の他、秘匿が必要な部分の暗号化、nonce を用いたリプレイ攻撃対策などの機能も備わっており、下位の転送方式に対してセキュリティの要求は課せられていない。CMP で実装が必須となっている暗号アルゴリズムは、署名用が DSA/SHA-1、MAC が RFC2510 で定める PasswordBasedMac、共通鍵暗号用が 3DES、鍵共有のための公開鍵アルゴリズムが Diffie-Hellman となっている。

Certificate Management Messages Over CMS (CMC)

CMC は CMP とほぼ同じ機能を実現しているが、CMP が新規にメッセージのフォーマットを規定しているのに対し、CMC は CRMF に加え、S/MIME で使われている PKCS #10(RFC2314) や Cryptographic Message Syntax (CMS, RFC3369) の標準を利用しているため、既にそれらの実装を行っているベンダにとっては、CMC 自体の実装は容易とされる。ただし、CMP には既に多くの実装があり、相互運用試験とその結果をふまえた仕様の改訂 (CMPv2) も進行中であるのに対し、CMC は相互運用試験もまだ行われておらず、普及に関して若干遅れているのが現状である。

CMC で用いる下位の転送方式としては、CMP 同様、MIME によるカプセル化を用いた e-mail や HTTP による転送、ファイルを用いた転送、およびソケットを用いた転送が規定されているが、いずれの場合も、下位の転送プロトコルに対してセキュリティに関する要求は課せられていない。CMC のメッセージは全て署名されており、オプションで nonce を用いたリプレイ攻撃対策も利用可能である他、メッセージの秘匿が必要な場合には、CMS で提供される暗号化の仕組みを用いてメッセージ全体が暗号化される。また、メッセージの秘匿に関しては、下位の転送プロトコルに SSL や TLS などを用いることで実現してもよいとしている。

証明書要求の際の POP に関しては、署名鍵の場合には要求者が証明書要求メッセージに署名を付加し、暗号鍵の場合には PKI と要求者の間で challenge & response を行い、Diffie-Hellman 鍵の場合には Diffie-Hellman Proof-of-Possession Algorithms (RFC 2875) を用いることとしている。

CMC では暗号アルゴリズムとして署名用に DSA(RSA の実装も推奨)、鍵交換用に Diffie-Hellman の実装が必須となっており、CMS ではメッセージの暗号化用アルゴリズムとして 3DES(RC2 CBC の実装も推奨) の実装が必須となっている。

3.3.3 Certificate Verification Protocols

CRL を用いて証明書の失効状態を確認する場合、証明書利用者は信頼パス上の CA の証明書に対するものも含む全ての CRL を利用者側で検査する必

要がある。CMP や CMC には証明書の失効処理に関する機能も備わっているが、失効状態の確認には CRL が用いられ、特定の証明書に関する失効状態をオンラインで問い合わせるための機能は提供されていない。

こうした背景から、特定の証明書に関して、その失効状態をオンラインで問い合わせるためのプロトコルが開発されている。

Online Certificate Status Protocol (OCSP)

OCSP は RFC2560 で規定されており、証明書の失効情報に関するリクエストとレスポンスからなる単純なプロトコルであり、下位に種々の転送プロトコルを用いることが可能であるが、もっともよく用いられているのが HTTP である。OCSP では問い合わせ対象の証明書を、証明書を発行した CA の名前と公開鍵のハッシュ値と、証明書のシリアル番号によって指定し、OCSP レスポンダは失効情報を “good”、“revoked” および “unknown” のいずれかの値で通知する。

OCSP のリクエストに対する応答はリアルタイムに得られるが、OCSP レスポンダが失効情報を取得する方法は実装に任されており、CRL を用いることも可能となっているため、必ずしも最新の失効情報が得られるとは限らない。

OCSP レスポンダは以下のいずれかに属する鍵を使ってレスポンスに対して署名を施し、応答の正当性を保証する。

- 問い合わせ対象の証明書を発行した CA
- 要求者が信頼する公開鍵を持っている OCSP レスポンダ
- CA より OCSP レスポンスの発行を許可されていることを示す証明書を持つ、CA 指定のレスポンダ (Authorized Responder)

署名やハッシュに用いるアルゴリズムはメッセージの中で OID によって指定され、実装がサポートすべき署名アルゴリズムは DSA(必須) と RSA(サポートすることを強く推奨) で、ハッシュ・アルゴリズムは SHA1 のサポートが必須となっている。また、リクエストにはオプションで nonce を含めることが可能であり、リプレイ攻撃に対する対処が必要な場合に利用できる。一方、メッセージの秘匿に関しては、OCSP 自身ではその機能を提供しないため、秘匿が必要な場合には SSL/TLS 等の下位レイヤーのサービスを利用することになる。

OCSP レスポンダは証明書の信頼パスの検証を行わないため、別途クライアント側で信頼パスを確立するのに必要な証明書の特定とその失効情報の確認を行う必要がある。また、ひとつのリクエストでひとつの証明書の失効情報しか問い合わせられない点も OCSP の欠点として指摘されている。このため、IETF pkix WG において、Simple Certificate Validation Protocol (SCVP)、

OCSPv2、Certificate Validation Protocol (CVP) 等のプロトコルが提案されているがいずれも現状ではドラフト段階となっている。

3.3.4 PKI 関連プロトコルの安全性について

PKI では CA に対する信頼をその安全性の根拠としているが、各 PKI 関連プロトコルにおいては、CA 公開鍵の PKI エンティティへの配布はプロトコル外の手段によって安全に行われることが前提とされ、プロトコルの規定対象外となっている。out-of-band での POP の確認などの手続きも含め、プロトコル外の手段によってなされるとされる手続きを規定するのは運用上のルールであり、このことから、PKI はその安全性の根幹部分を運用に大きく依存していると言える。このため、プロトコル自身の安全性もさることながら、プロトコルで規定されていない部分の運用ルールとチェック機構の確立こそが、PKI 運用における安全性の確保に重要であると言える。

3.4 IC カード・プロトコル

現在の IC カードのプロトコルは、ISO/IEC 7816 で定義された仕様をもとに実装されている。その中で、通信プロトコルに関するセキュリティ機能は、7816-4 に記述されている。従って、以下に 7816-4 のセキュリティ機能について概説し、その後通信プロトコルの安全性を評価・報告する。

3.4.1 ISO/IEC7816-4 プロトコルのセキュリティ機能

IC カードのプロトコルは、IC カードへのコマンド送信と、IC カードからのレスポンスからなる。コマンドは、Command Header (CH) と、Command Body(CB) から構成され、レスポンスは、Response Body(RB) と、Response Trailer(RT) から構成される。さらに、CH は、コマンドのストラクチャを定義する Class Byte(CLA)、コマンドの内容を表す Instruction Byte(INS)、と２つの付加パラメータ（P1, P2）から、CB は、データフィールドのバイト長を表す Lc field、データが格納される Data field、レスポンスの最大バイト長を規定する Le field から構成される。一方 RB は、Data field から、RT は、SW1、SW2 という２つの Status byte から構成される。図 16 にデータ構造と、データ長を示す。

CLA の下位４ビット（b4-b1）の内、b2,b1 の２ビットで Logical Channel Number を定義し、b4 ビットで当該コマンドメッセージが、セキュリティメッセージであるか否か、b3 ビットで CH がメッセージ認証対象データになっているかを定義する。

上記プロトコルに対して、7816-4 で定義しているセキュリティ機能は以下

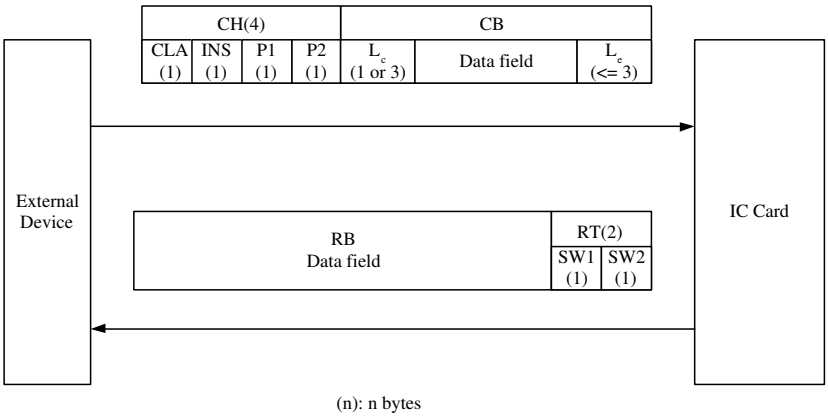


図 16: IC カードのプロトコル構造

の 2 つである。

- メッセージ認証
- 暗号化

メッセージ認証に関しては、共通鍵暗号アルゴリズムを使用した Cryptographic checksum と、公開鍵暗号アルゴリズムを使用した署名の 2 種類が定義されている。暗号化は、共通鍵暗号アルゴリズムを使用する。鍵については、秘密鍵をそのまま使用するケースと、セションキーを導出し、使用するケースと 2 通りがあり、Auxiliary security data object（後述）を使用して、定義できる。暗号アルゴリズム及び、セションキーの導出方法に関する具体的な指定はない。

Data field には、Tag(T), Length(L), Plain Value(V) の 3 要素からなる Data object が格納される。コマンド及びレスポンス内に格納される Data object の構成を以下に示す。

Security Messaging においては、以下の 3 つの Data object が定義されている。

- plain value data object〜平文データのオブジェクト
- security mechanism data object〜セキュリティ機能の演算結果を格納するオブジェクト
- auxiliary security data object〜セキュリティ機能に関するパラメータを格納するオブジェクト

メッセージ認証における Cryptographic checksum(CC) は、ISO/IEC9797 で定義されているいわゆる CBC-Mac 方式により演算される。Initial check

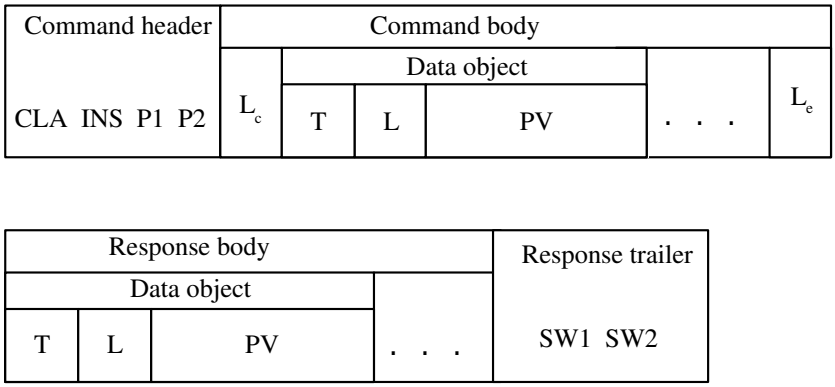


図 17: データオブジェクトの構成

block には、(a)null、(b) 直前のプロトコルの CC、(c)initial value block、(d)auxiliary data のブロック、のどれかが使用される。Cryptographic checksum のバイト長は 4 バイト以上と定められている。CC は、security mechanism data object である Cryptographic checksum data object（Tcc ||Lcc||CC で構成）に格納される。Command におけるメッセージ認証対象データは、他のオブジェクトの T, L, PV であり、T の最下位ビットが 1 である場合、オブジェクト全体がメッセージ認証対象データとなる。また、設定によって CH も認証対象データに含まれる。Response における認証対象データは、T の b1 ビットが 1 である他のオブジェクトの T, L, PV および、RT（SW1 及び SW2）である。

署名データは、Digital signature input data と、Digital signature の 2 種類のデータを Digital signature related data object から構成される。証明対象データは、CC と同様であり、メッセージリカバリーが可能な証明とするか否かの 2 種類のアルゴリズムの利用が想定されている。

暗号化におけるデータオブジェクトは、Data object for confidentiality というオブジェクトを使用する。格納する情報は、暗号化されたコンテンツである Cryptogram(CG) と、パディング情報である Padding Indicator(PI) の 2 種類である。PI は、データが BER-TLV でコーディングされている場合は、不要となる。暗号化する複数のデータオブジェクトは、暗号化され CG にカプセル化される。Data object for confidentiality は、パディング情報の有無で、TPI,CG||LPI,CG||PI||CG、もしくは TCG||LCG||CG という構造となる。

auxiliary security data object は、アルゴリズム、鍵、Initial check block などの設定情報を格納し、様々なセキュリティパラメータの交換に使用される。

3.4.2 安全性の評価

攻撃者の想定

IC カードのプロトコルのセキュリティ機能は、IC カード内に保管されている秘密鍵を元に構成されている。事前共有型の共通鍵アルゴリズムを利用するため、外部機器にも同様の秘密鍵が保管されている可能性がある。攻撃方法としては、外部機器を不正操作することによって IC カードにアクセスする方法と、外部機器-IC カード間の通信プロトコルに対して攻撃を行なう場合の 2 通りが想定される。外部機器に対する攻撃については、プロトコルの安全性を検証しようとする本稿の目的からは外れるため、特に議論を行なわない。すなわち、外部機器は、何らかの利用者認証手段を備えており、攻撃者によって不正操作されることがないことを前提とする。よって安全性の検証の前提条件として、以下の仮定を置く。

- 攻撃者は、通信路に対して攻撃を行なう。
- 攻撃者は、秘密鍵を事前に入手できない。

様々な攻撃法に対する検証

- 不正メッセージ

  秘密鍵をもつエンティティのみが、セキュリティオブジェクトを使用したメッセージを作成できる。従って、IC カードや外部機器がセキュリティメッセージのみを許容したいた場合には、秘密鍵を持たない不正者が不正メッセージを受信させるような攻撃は行なえない。

  しかし、セキュリティメッセージ、通常のメッセージ両方を許容するような設定になっていた場合には、攻撃が成功する可能性がある。

- メッセージの改ざん

  7816-4 では、メッセージの一部分を任意に変更するような改ざん攻撃に対する対策として、メッセージ認証機能を提供している。以下、メッセージ認証に使用するアルゴリズムは安全である、という前提のもとに検証を進める。メッセージ認証機能が使用されていた場合、Command メッセージについては、Lc、Le 以外の全てのデータをメッセージ認証の対象データとすることが可能であり、改ざんはできない。Response メッセージの全ては、メッセージ認証の対象データとすることができるため、やはり改ざんは不可能である。また、以前の通信プロトコルから収集したデータを使用して改ざんを行なうカットアンドペースト攻撃なども行なうことはできない。

  脅威として想定されるのは、Lc に対する改ざん攻撃である。Lc を改ざんすることで、Command Body 長を拡大し、任意のメッセージオブジェクトを挿入できる可能性がある。また、Response Body に対しても、Le の範囲内において任意のメッセージオブジェクトを挿入するような攻撃が想定される。あるメッセージオブジェクトが、メッセージ認

証対象データか否かは、オブジェクトのタグ情報 (T) によって設定するため、挿入可能であると考えられる。従って、挿入されたメッセージオブジェクトを受信エンティティで排除するように設定を行なう必要がある。CH は、メッセージ認証対象データとするか否かを選択可能となっているが、改ざんによってセキュリティ機能を無効化できるため、必ず対象データとすべきである。ただし、セキュリティ機能を無効化した場合でも、受信エンティティが、セキュリティ機能なしのメッセージを許容しないように設定されていれば、脅威とはならない。

- メッセージの盗聴

  メッセージの盗聴に対しては、データの暗号化機能を提供している。従って、暗号化された対象データを復号化し盗聴することは、暗号アルゴリズムが安全であるという前提のもとでは不可能であると考えられる。また暗号化機能は、カットアンドペースト攻撃などを考慮し、メッセージ認証と組み合わせて使用するべきであると考えられる。

- 再送攻撃

  再送攻撃とは、以前の通信で収集したメッセージや、他の利用者は送受信したメッセージをそのまま送信するような攻撃である。7816-4 では、再送攻撃に対する対策として、メッセージ認証子を作成する際の Initial block に直前の通信で使用したメッセージ認証子を使用する方式が定義されており、この方式を使用すれば再送攻撃は成功しないと考えられる。しかしながら、最初から最後までのプロトコル全てを再現するような再送攻撃は（それがどの程度有効な攻撃であるかはわからないが）実行できる可能性がある。

まとめ

IC カードのプロトコルは、以下のような点に留意し、メッセージ認証及び暗号化機能を適切に使用すれば、安全であると考えられる。

- 暗号アルゴリズムは、安全なアルゴリズムを選択する。
- 通信路が攻撃者によって攻撃される可能性がある場合には、各エンティティは、すべてのデータオブジェクトに関してメッセージ認証されたメッセージ以外は許容しないように設定する。
- CH は、必ずメッセージ認証対象データとする。
- Cryptographic checksum は十分な長さのものを使用する。
- メッセージ認証の chaining mode は再送攻撃などを考慮し、利用することが望ましい。
- 重要な情報については、暗号化を行なう。

4　暗号プロトコル技術マップの検討

２章における電子政府で用いられる暗号プロトコルに関する調査結果に基づき、電子政府で用いられる暗号プロトコルのリストを整理するとともに、将来の電子政府として考慮すべき暗号プロトコルの技術動向について考察する。

4.1　電子政府システムで利用される暗号プロトコルリスト

(1) SSL
電子政府システムにおいて、もっとも利用されている標準技術に基づく暗号プロトコルとして SSL が挙げられる。SSL は、公開鍵暗号に基づき、鍵共有、相互認証、情報秘匿、データ完全性機能を提供する暗号プロトコルであり、主な電子政府システムの要件は、SSL を用いることで保証されるケースが多い。また、SSL は、TCP/IP 上のセッション層で稼動するプロトコルであり、下位レイヤでセキュリティ機能を実現する IPsec と異なる。SSL は最も安全性が高い SSLv3 が用いられている。

(2) S/MIME
S/MIME は、情報秘匿、鍵配送、および、否認防止機能を提供するセキュアメッセージ技術である。利用者への情報通知の手段として S/MIME に従う電子メールが用いられる場合がある。例えば、入札システムにおける、入札結果を入札者に通知する場合などである。このような場合には、開札情報の偽造や、入札業者に関する情報が漏洩するなどの脅威を考慮して、S/MIME などのセキュアな電文を送受する必要がある。

(3) XML セキュリティ
XML は、情報秘匿および電子署名 (否認防止) 機能を提供するセキュアメッセージ技術である。電子申請のフォーマットを共通化し、申請書の各属性をオブジェクト化して扱う場合に XML が利用される。MXL では、オブジェクト単位で、暗号化や署名を行う技術として XML セキュリティ技術が用いられる。具体的には、国税電子申告・納税システムなどで用いられている。

(4) IPsec
IPsec は、鍵共有、情報秘匿、データ完全性、相互認証機能を実現する暗号プロトコルであり、IP ネットワークを用いて閉域網を形成するすなわち、IP/VPN を実現する技術として利用される。例えば、電子政府のサブネットワークを IP 網で構成する場合（マルチペイメントネットワークの場合など）や、中央省庁と自治体を接続するＬＧＷＡＮなどでハードウェア回線暗号としての利用が想定される。

(5) IC カードプロトコル

各種電子政府システムにおける、利用者（クライアント）側のセキュリティ向上や利便性の確保を目的として、主な電子政府システムが IC カードの利用を想定している。

IC カードの暗号プロトコルとしては、IC カードホスト間の伝送プロトコルとして、ISO/IEC 7816-4(接触型) や ISO/IEC 14443-4（非接触型）の利用が必須となる。本伝送プロトコルでは、情報秘匿および、データ完全性機能が提供される。　特に、IC カード普及の鍵を握るシステムとして住基カードがある。住基カードとしては、近接型の非接触ＩＣカード (ISO/IEC 14443) の利用が想定されている。

(6) PKI
政府認証基盤を実現する技術として、各種公開鍵とその所有者を紐付ける PKI は非常に重要な位置を占める。基本的には電子証明書として X.509v3、失効リストは X.509 CRLv2 、を、また、リポジトリプロトコルとして LDAPv3 が用いられる。運用管理プロトコルの標準がＩＥＴＦなどで検討されているが、現状の政府認証基盤では、利用についての規定がない。

4.2　技術マップ

電子政府システムが発展していくに従い、現状の暗号プロトコル技術に基づき設計が行われているシステムは、最新の技術を反映していく必要が生じる。特に、セキュリティ技術の進歩は、解読や攻撃の進歩と表裏一体の関係にあり、設計時に想定されなかった脆弱性が、後に重大な脅威となり得るため、信頼性の高いシステムを運営していくには、常に最新のセキュリティ技術を反映していくことが望ましい。ここでは、現在使われている暗号プロトコルに関する方向性について考察し、電子政府システムで利用される暗号プロトコルの技術マップを検討する。

(1) ＳＳＬについて
SSL は、プロトコル階層構造において、セションレイヤに位置するプロトコルである。すなわち、実装の観点からは、アプリケーションが SSL のセッションを管理するため、アプリケーションがそれぞれ SSL 対応に改修する必要がある。従って、セキュリティの観点からは、SSL のプロトコルに処理異常が発生した場合には、アプリケーションがセッション状態を管理できるため、その異常を把握し対処を明確に規定できる点で、下位レイヤでセキュリティ機能を提供する IPsec よりもセキュリティ管理が容易であると言える。すなわち、実装と安全性の間にトレードオフがあり、SSL のような上位層でのセキュリティ機能は、エンドーエンドでセキュリティ機能を提供し、安全性を重視したシステムを実現

するのに適していると考えられる。

SSL が提供するセキュリティ機能を考慮すると、今後も SSL プロトコルが電子政府システムで利用される傾向は続いていくと思われる。SSL は、暗号アルゴリズムを複数選択可能となっているため、ある暗号アルゴリズムに脆弱性が発見されたとしても、他の安全な暗号アルゴリズムに置換し使用するという手法が考えられる。従って、暗号アルゴリズムについては、柔軟性のあるプロトコルであるといえる。しかしながら、暗号プロトコル攻撃手法の発展にともない、現 SSL が将来に渡って安全でありつづける保障はない。しかも、SSL は、現在完全に仕様が凍結されており、IETF では SSL とほぼ同等のセキュリティ機能を有する TLS に関する仕様拡張を進めている。

従って、今後の電子政府システムは、SSL と同等のセキュリティ機能を有し、種々の改善が施され、現状の最新技術が反映される TLS を利用するべきであるといえる。

(2) 蓄積メッセージの暗号プロトコルについて
さまざまなエンティティを介する電子政府システムのアプリケーションにおいて、エンドエンドで、情報秘匿や否認防止を実現する技術として S/MIME や XML セキュリティ技術は、今後も必要となる技術である。しかしながら、業務アプリケーションが複雑になりワークフローなどを電子的に自動化していくことを考慮すると、オブジェクトを任意に設定でき、オブジェクト単位で暗号化や署名などのセキュリティ機能が実現できる XML セキュリティ技術のほうが、S/MIME より発展性があると考えられる。また、昨今、メールクライアントや Web ブラウザが XML の処理機能を具備しつつある。従って、当面は、S/MIME と XML が併用されるが、今後、XML の処理系がインターネットの標準技術として発展し、現在進行中の XML セキュリティの標準化が進むとともに、情報フォーマットのセキュリティ技術としては、XML 技術が今後使われていくことが予想される。

(3) ＰＫＩについて
PKI については、現状、PKI 技術に準拠した電子証明書や失効リストの発行管理を政府認証基盤で実現している。今後は、発行や失効、有効性確認などをオンラインで実現していく可能性があり、現在 IETF で検討が盛んに行われている証明書の管理運用プロトコル CMP(Certificate Management Protocol),CMC(Certificate Management Message over CMS),OCSP(On-line Certificate Status Protocol),SCVP(Simple Certificate Validation Protocol),CVP(Certificate Validation Protocol) などの標準プロトコルの利用が予想される。また、柔軟なアクセス制御を行なうことが可能な PMI（Privilege Management Infrastructure）が、

ＰＫＣと連携利用される属性証明書として普及していくことが予想される。

(4) IPsec について
IPsec については、アプリケーションに依存しない下位レイヤでのセキュリティ機能の位置付けで、VPN を実現する技術として引き続き利用されるものと思われる。なお、IPsec についても、暗号アルゴリズムは選択的に利用可能となっている。

(5) IC カードについて
IC カード関連については、利用クライアント数が膨大であることや、一度発行すると５年程度は利用されることを想定すると、物理的な伝送速度の向上は見込まれるが、ISO/ IEC 7816 や ISO/IEC14443 レベルの伝送プロトコルの基本仕様は、大きな問題が無い限り当面利用されつづけるものと思われる。今後、一枚の IC カードで複数の業務アプリケーションに動的に対応させるなどの必要があるので、このような IC カードのマルチアプリケーション対応に関して、アプリケーションのダウンロードやアクティベーションの実現手法に関する、統一的な仕様が必要になると考えられる。

(6) 独自プロトコルについて
独自プロトコルは、システム要件に従って、個々にカスタマイズされたセキュリティプロトコルと考えられ、いくつかの電子政府システムでも用いられている。特殊なシステム要件や制約がある場合には、独自プロトコルを用いることのメリットがあると考えられる。独自プロトコルは、仕様が非公開であることにある程度の安全性の根拠をおいているが、その反面、専門家のスクリーニングを得ていない点で、様々な観点での脆弱性が潜在している可能性もある。従って、特殊な用途や制約で標準的なプロトコルの利用が不可能な場合を除いては、独自プロトコルを用いないのが望ましいと考えられる。あえて、独自プロトコルを用いる場合には、設計、実装、運用それぞれのフェーズにおいて、セキュリティに関して十分な見識を有する開発者、運用者が従事することが望ましい。

電子政府で用いられる暗号のリストと今後の技術動向を表 12 にまとめた。

5　暗号プロトコル安全性評価手法の検討

5.1　証明可能安全性付き暗号プロトコルの研究の流れ

暗号プロトコルとは、鍵共有 (配送)、相手認証などのセキュリティプロトコルを指す。暗号プロトコルについては、例えば、Kerberos, SSL/TLS, SSH,

	提供機能	**プロトコルレイヤ**	今後の予想される動向
SSL	相手認証、鍵交換、情報秘匿、データ完全性	セッション	仕様が凍結していることもあり、TLS に置換される。
S/MIME	鍵交換、情報秘匿、データ完全性	アプリケーション	オブジェクトレベルの柔軟な機能を提供する XML に置換される。
XML セキュリティ	情報秘匿、否認防止（署名）	アプリケーション	ブラウザやメーラなどに標準機能として搭載されつつある。
PKI		アプリケーション	有効性確認や、発行処理などの管理プロトコルの利用、認証対象の多様化（属性認証など）など
IPsec	相手認証、鍵交換、情報秘匿、データ完全性	IP レイヤ	アプリケーションに依存しない、ハードウェア回線暗号装置として、ルータなどで搭載され引き続き利用される。
IC カード	情報秘匿、データ完全性	ＩＣカード、ホスト間のデータリンクレイヤ	伝送プロトコルは当面利用される。マルチアプリケーション技術の発展が期待される。
独自プロトコル	任意	任意	標準技術を使うことが望ましいが、要件や制約から独自プロトコルが用いられる場合もある。

表 12: 暗号プロトコルリスト

IPSec など様々な方式が提案されており、その目的（守るべきセキュリティの要件）、利用環境（計算機の能力や、通信帯域）などによって、それぞれカスタマイズされてきた経緯がある。これらの暗号プロトコルの設計においては、各設計者がもつ、セキュリティ、暗号技術の見識に基づき、ヒューリスティックなアプローチで経験的に安全な方式が提案されてきた。しかしながら、それらのプロトコルが、提案され、仕様が公開された後に、セキュリティ上の脆弱性が指摘されるケースが散見されるのが実状である。実際、標準化団体（例えば、IETF など）で標準化された暗号プロトコルが、製品等を通じて広く普及した後、脆弱性が発見される場合など改修に関わるコストが問題となる。特に、致命的な脆弱性が発見された場合には、経済的な損害の回避を目的とし、早急に対処（ソフトウェアではパッチを充当する。ハードウェアではチップや部品を交換する）する必要が生じる。従って、その技術や製品が普及すればするほど、社会的なインパクトは非常に大きくなる。このような背景を鑑み、従来から、暗号プロトコルの安全性を客観的に評価する手法の必要性が求められてきた。Needham, Schroeder らは、その重要性について当初から指摘している [11, 12]。

暗号プロトコルの安全性を評価する技術として、Burrows, Abadi, Needham らは、特殊なロジックを用いて、解析するプロトコルの問題を指摘する方法を提案した（いわゆる BAN ロジック）[13]。BAN ロジックは、論理体系に基づくフォーマルなプロトコル解析・検証手法である。しかしながら、これらのアプローチは、プロトコルの論理的不備を発見する目的では、有効で

あるが、ＢＡＮロジックで用いる抽象的なオペレーション（たとえば、ディジタル署名といったオペレーション）などを実際の方式（例えば、RSA 署名）に適用した場合の安全性を保障するものではないことが指摘されている。

このような経緯および、計算量的な観点からの暗号方式の証明可能安全性の技術を応用し、暗号プロトコルの安全性を計算量的な観点から証明するアプローチが Bellare, Rogaway によって、1993 年に初めて考案された。Bellare らは、現実性のある暗号プリミティブ（擬似乱数生成器）の存在を仮定した場合に、計算量的に安全性が証明可能な相互認証プロトコルおよびセッション鍵の共有プロトコルが実存することを示した。

具体的には、２者間の相互認証プロトコルとして、2 者 (i, j) 間で、共有できる鍵を生成するジェネレータ g( ), プロトコルのすべてを制御でき、多項式時間の計算機能力を有する攻撃者 E、相互認証する２者を模擬するオラクル $\Pi^s_{i,j}$（セッション s において、プレーヤ i が、プレーヤ j を認証するオラクル）、および、$\Pi^t_{j,i}$、を定義し、攻撃者がこれらのオラクルと対話する場合において、安全な相互認証プロトコルは、以下のように定義される。

1. (Matching conversations ⇒ acceptance)：上記の２つのオラクルが、相互認証プロトコルを実行の後、それぞれ受理状態になる。

2. (Acceptance ⇒ Matching Conversations)：起動側のオラクル $\Pi^s_{i,j}$ が受理状態になり、かつ、対応する応答側のオラクル $\Pi^t_{j,i}$ が存在しない確率が、無視できるほど小さい。

その安全性証明可能な具体的なプロトコルとして、例えば、f が擬似乱数生成関数の属であるという仮定のもと、以下を提案している。

$$\text{A a} \quad \xrightarrow{\qquad R_A \qquad} \quad \text{B a}$$
$$\xleftarrow{\qquad [B|A|R_A|R_B]_a \qquad}$$
$$\xrightarrow{\qquad [A|R_B]_a \qquad}$$

ここで、a：事前共有鍵、R_A, R_B ：A， Bそれぞれのエンティティで生成される乱数、[x]a：(x, fa(x))、f：擬似乱数関数の属

また、証明可能な安全性を有する認証付き鍵交換プロトコルは以下のように定義される。

1. (Benign adversary ⇒ keys according to Sk)：2 つのオラクルに対して、問合わせを行ない、その結果を忠実に相手のオラクルの入力とする忠実な攻撃者を仮定した場合、２つのオラクルは、必ず、Sk の分散に従った同一値 α (共有鍵) を出力して受理状態になる。

2. (Session key is protected)：任意の多項式時間計算能力を有する攻撃者に対して、Test Query(オラクルがコイントスを行ない、１の場合はセッ

ション鍵αを、0の場合はSkから乱数を選択して、応答する問合せ) において、その結果から攻撃者がコイントスの値を有意性をもって推測する確率が無視できるほど小さい。すなわち、
$advantageE(k) = max\{0, Pr[Good - GuessE(k)] - 1/2\}$ is negligible

その安全性証明可能なプロトコルとして、例えば、f, f ’が擬似乱数生成関数の属であるという仮定のもと、以下を提案している。

$$\text{A} \quad \xrightarrow{\qquad R_A \qquad} \quad \text{B a1,a2}$$
$$\xleftarrow{\quad [B|A|R_A|R_B|\{\alpha\}_{a2}]_{a1} \quad}$$
$$\xrightarrow{\qquad [A|R_B]_{a1} \qquad}$$

ここで、a1,a2：事前共有鍵、R_A, R_B：A，Bそれぞれのエンティティで生成される乱数、[x]a：(x, fa(x))、{ α } a2：(r, f ’a2(r) + α)、 f,f ：擬似乱数関数の属

以後、Bellareらを中心として、鍵共有プロトコルに関して、証明可能安全性に関するプロトコルが提案されてきた。例えば、公開鍵暗号に基づく、2者間のプロトコルとして、[14] などがあり、また信頼できる仲介者を想定した3者間のプロトコルとして、[15, 16] らがある。

一方、安全性については、その後、さまざまな攻撃が定義され、それに耐性を有するより強力な証明可能なプロトコルが提案されている。以下は、これまで定義されている安全性のモデルについて述べる。その意味で上述のプロトコルは、後述の既知鍵攻撃安全および意味論的安全を保障しているに過ぎない。

5.1.1 既知鍵攻撃安全 (Secure against known key attack)

ある鍵共有プロトコルにおいて、攻撃者が1つのセッション鍵を得ることができたとしても、他のセッション鍵を得ることができないとき、そのプロトコルは、既知鍵攻撃安全であると定義する。

Needham, Schroederのプロトコルを例にとる。

(1) $A \to S : A, B, N_A$

(2) $S \to A : \{N_A, B, K, \{K, A\}_b\}_a$

(3) $A \to B : \{K, A\}_b$

(4) $B \to A : \{N_B\}_K$

(5) $A \to B : \{N_B - 1\}_K$

(1)〜(3) が、鍵共有プロトコルであり、(4)〜(5) は、B が A を認証するプロトコルである。

ここで、もし、攻撃者が過去の通信履歴を記録しておき、あるときそのセッション鍵 K が攻撃者に破られたとすると、その攻撃者はAに代わって、記録しておいた通信履歴を用いて (3) において B に対してリプレイ攻撃をかけると、B は、鍵がAによって新たに K に更新されたと解釈し、以後のセッション鍵を K に設定する、いわゆる良く知られた Denning-Sacco の攻撃がこれに相当する。すなわち、1つのセッション鍵を得ることにより、攻撃者は能動的に以後のセッション鍵を通信履歴の再利用という制約下で制御できることになる。

Bellare らは、enc,mac が安全な暗号プリミティブであるとの仮定のもと、既知鍵攻撃安全な 3 者間の具体的な鍵共有プロトコルとして以下を提案している。

(1) $A \rightarrow B : R_A$

(2) $B \rightarrow S : R_A, R_B$

(3) $S \rightarrow A : enc_{a[1]}(K), mac_{a[2]}(A, B, R_A, enc_{a[1]}(K))$

(4) $S \rightarrow B : enc_{b[1]}(K), mac_{b[2]}(A, B, R_B, enc_{b[1]}(K))$

5.1.2 意味論的安全 (Semantic Secure of session key)

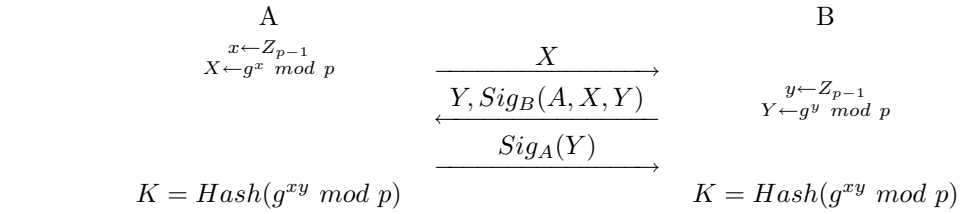
暗号学的な定義同様に、攻撃者がセッション鍵のいかなる部分的な情報も入手不可能であるとき、そのプロトコルを意味論的安全（ここでは、あえて、暗号学の強秘匿と異なる用語を用いる）であると定義する。上記の Bellare らの認証付き鍵共有プロトコルは、本安全性を満足している。

5.1.3 Forward Secrecy

鍵共有プロトコルが実行され、セッション鍵Kの共有が完了した後、攻撃者が各エンティティの秘密鍵（あるいは事前共通鍵）を得ることができたとしても、セッション鍵Kのいかなる情報も得ることができないとき、そのプロトコルは Forward Secrecy であると定義する。

この仮定は、通信の盗聴や能動的な攻撃よりも、エンティティに相当するクライアントやサーバをクラックして、そこに保管されている秘密鍵を得られる場合が実際の攻撃としてあり得るという仮定に基づいて、クラックが発生した場合でも、通信の履歴をすべてモニターしていない限り、その攻撃者は、鍵情報を得ることができない安全性に関する。

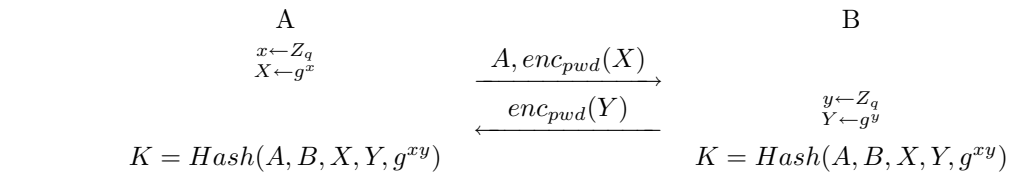
Sigi、Hash、DH の暗号プリミティブの安全性の仮定のもと、具体的なプロトコルとして、既知鍵攻撃安全、かつ、意味論的安全、かつ、Forward Secrecy な認証付き鍵共有プロトコルを提案している。



5.1.4 辞書攻撃安全

オフライン検索によって除去されるパスワードの候補の数が、高々、攻撃者によって保持できるプロトコルの通信履歴の数である時、そのプロトコルを辞書攻撃安全と定義する。概念的に本攻撃は、いわゆる辞書攻撃によって、パスワードが推測されることによる攻撃が不可能であるとき、そのプロトコルは辞書攻撃安全であるとする。

enc、Hash、DH の暗号プリミティブの安全性の仮定のもと、具体的なプロトコルとして、既知鍵攻撃安全、かつ、意味論的安全、かつ、Forward Secrecy、かつ、辞書攻撃安全な鍵共有プロトコルを提案している。



5.1.5 まとめ

ここでは、暗号プロトコルの安全性を計算量的な観点から証明する手法について、研究の流れをまとめた。これらの視点を総括すると、暗号プロトコルの設計においては、セキュリティに関する分析がやりやすい、プロトコルにおけるそれぞれのデータフローの目的が明確である、効率的である、などの視点を常にもちながら行う必要がある。さらに、客観的な安全性の指標として証明可能であることが重要であり、冒頭で述べたような潜在する脆弱性をなくするために、新たなプロトコルを設計する際には、これまで定義されてきた安全性の定義に基づく証明可能安全性が保証されているべきである。また、新たな試みとして、証明可能安全性とフォーマル検証を融合した安全性評価のアプローチも行われている [17, 18]。

5.2 暗号プロトコル安全性評価手法に関する考察

以下では、種々の暗号プロトコルに関する既知の脆弱性に対して大まかな分類を試み、その分類に基づき、安全性を評価するための手法や方針についての検討を行う。

3 節で述べた各種プロトコルの脆弱性は、実装上の原因によるものを除くと、その脆弱性が何に起因するかによって以下のように分類できると考えられる。

- 暗号方式上の脆弱性
  DES に対する線形解読、PKCS #1-v1.5 に対する million message attack、PKCS #1-v2.0 OAEP に対する選択暗号文攻撃、DSA における乱数生成の偏りなど
- 鍵確立/鍵交換、認証等に用いるプロトコルの脆弱性
  man-in-the-middle attack, replay attack, known-key attack, DH に対する small-subgroup attack などへの脆弱性
- 暗号アルゴリズム/プロトコルの不適切な使用
  DES を用いる IPsec ESP の仕様における IV に対する予測可能性の許容
- 運用上の脆弱性
  安全性の低いオプションの許容 (強度が十分でない暗号アルゴリズムの利用、IPsec における認証なしの ESP の利用)、仕様が意図するセキュリティ上の目的と利用者側の認識とのずれ (IKE における、なりすましを行う攻撃者への identity の漏洩、蓄積メッセージの不正転送) など

このように見た場合、三点目までは評価対象のプロトコルが building block として利用する基本的な暗号アルゴリズムや鍵交換などのプロトコルと、その利用法に関する安全性の問題である。このため、暗号プロトコルの安全性評価にあたっては、5.1 節で述べた証明可能安全性に関する研究の成果も踏まえ、まず評価対象プロトコルが利用する基本的な暗号アルゴリズムやプロトコルが、その安全性について十分に検証されたものであるか否か、また、それらが既知の脆弱性を抱えている場合には、その脆弱性が問題となるような利用形態を許しているのか否かを確認することが必要である。現在のところ形式的な手法は適用性に制限があり、IKE のように、仕様が複雑で形式的な検証が困難なプロトコルもあるため、現実的には、既知の攻撃に対する安全性を逐一検証する ad hoc な手法に頼らざるを得ない場合が多いと考えられる。そうした場合、既知の攻撃に関する詳細なデータベースを構築することなどにより、ad hoc な方法での安全性の検証をサポートする仕組みの整備が重要と考えられる。

こうしたプロトコル仕様上の基本的な安全性に関する検証は、仕様策定時に十分行われているとみなすことも可能であるが、四点目の、運用上生じる

可能性のある脆弱性に関しては、利用目的や利用環境に関して具体的な情報を持つ利用者の立場でのみ詳細な検討が可能であると言え、利用者側で十分な検証を行う必要がある。一般にプロトコル仕様は広汎な利用形態や互換性などを考慮し、種々のオプションを備えている場合が多く、利用形態によっては、安全性の観点からその使用が適切でないオプションが存在する場合がある。このため、仕様で利用を許容されていることが、安全性を保証されていることと等価ではないことに注意し、利用形態に即した適切なプロトコルの使用法と、その場合の安全性に関する評価を行うことが必要である。3 節で述べた脆弱性の例からは、利用形態に応じてその利用の妥当性や安全性を判断する必要のあるものとして以下のような事項が挙げられる。

- 安全性の低いオプションの使用の是非
  IPsec で完全性チェックを伴わない ESP を使用する場合には上位層での完全性チェックが前提とされおり、単独での使用は適切でない。

- 種々のオプションと提供されるセキュリティ上の機能の相関の把握
  IKE のメインモードにおいて公開鍵署名を用いた場合、IKE セッションの initiator 側のノードの identity はなりすましを行う攻撃者にに対して秘匿されないが、このように、使用するオプションと提供されるセキュリティ上の機能の相関が自明でなく、適切なオプションの選択が困難な場合がある。あるオプションによってどのような機能が提供される/されないのかを正確に把握し、利用目的に合わない、誤ったオプションの選択が利用者などによってなされないような対策や注意が必要である。

- 既知の脆弱性に対する対処の要否
  S/MIME における million message attack 対策の要否が、メッセージの復号が自動的に行われるか否かによって決まり、small-subgroup attack 対策の要否が、使用する鍵ペアの種別や out-of-band での復号の成否の伝達の可能性によって決まるように、既知の脆弱性が現実的な問題となるか否かは利用形態に即して判断し評価する必要がある。

以上を考慮すると、暗号プロトコルの評価手法はおおまかに以下のようにまとめることができる。

1. 暗号プロトコル中で利用する暗号技術の安全性の検証
   - 既知の攻撃に対する安全性の ad hoc な検証
   - BAN ロジック等を用いた形式的検証
   - 計算量理論に基づく証明可能安全性の検証
2. 暗号プロトコル全体の分析

- 暗号プロトコルが実現するセキュリティ上の機能の明確化
  (動作モード、オプション、その他の条件などに応じて)
- 既知の脆弱性の調査 (条件、対処法)

3. 利用者側での運用形態の分析

- セキュリティに関する要求条件の分析
- 同時に利用する他のセキュリティ技術の有無や相関
  (ネットワーク層、トランスポート層、アプリケーション層など)
- 実装がサポートするアルゴリズムやオプションの調査

4. 上記 2. と 3. の適合性の検証

- 利用目的に合致する適正な動作モード／オプションの有無
- 管理者／利用者が正しく暗号プロトコルを使用するためのサポートの要否

6 むすび

本調査では、電子政府システムで既に利用されている暗号プロトコルの標準技術についてリストアップするとともに、これらの暗号プロトコルの安全性評価について、プロトコル仕様に内包する脆弱性、暗号アルゴリズム/暗号プロトコルの不適切な使用による脆弱性、運用上の脆弱性の観点から調査を行なうとともに、電子政府システムで用いられる暗号プロトコルの今後の動向について考察を行なった。また、暗号プロトコルの評価の一手法となりえる証明可能安全性な暗号プロトコルに関する研究の動向を調査した。

これらの調査結果から得られた知見をもとに、暗号プロトコルを評価するにあたり考慮すべき点を提言として最後にまとめた。本調査報告が暗号プロトコルの客観的な評価手法検討の一助となれば幸いである。

参考文献

[1] N. Ferguson, B. Schneier: “A Cryptographic Evaluation of IPSec”, http://www.counterpane.com/ipsec.html, 1999.

[2] S. Bellovin: “Problem Areas for the IP Security Protocols,” Proc. Sixth Usenix Security Symp., Usenix Assoc., Berkeley, Calif., 1996

[3] S. Vaarala, A. Nuopponen, T. Virtanen: “Attacking Predictable IPsec ESP Initialization Vectors” ICICS 2002: 160-172, 2002

[4] R. J. Perlman, C. Kaufman: “Key Exchange in IPSec: Analysis of IKE”, IEEE Internet Computing 4(6): 50-56, 2000

[5] W. A. Simpson: “IKE/ISAKMP Considered Harmful”, USENIX ;login:, December 1999

[6] M. Abadi, R. Needham, “Prudent Engineering Practice for Cryptographic Protocols”, Digital SRC Research Report #125, June 1994

[7] A. Anderson, R. Needham: “Robustness Principles for Public Key Protocols”, in LNCS 963, Advances in Cryptology Crypto ’95, pp.236-247. Springer-Verlag, 1995

[8] D. Davis: “Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML”, in Proc. Usenix Tech. Conf. 2001, Boston. June 25-30, 2001.

[9] C.H. Lim, P.J. Lee: “A key recovery attack on discrete log- based schemes using a prime order subgroup”, B.S. Kaliski, Jr., editor, Advances in Cryptology - Crypto ’97, Lecture Notes in Computer Science, vol. 1295, 1997, Springer-Verlag, pp. 249-263.

[10] IEEE P1363, Standard Specifications for Public Key Cryptography, 1998, work in progress.

[11] R. Needham, M.Schroeder: “Authentication Revisited”, ACM Operating Systems Review, Vol.21, No.1, 1987

[12] L. Gong, R. Needham, R. Yahalom: “Reasoning about belief in cryptographic protocols”, in IEEE Computer Society Symposium in Security and Privacy, pages 234-248. IEEE Computer Society Press, May 1990

[13] M. Burrows, M. Abadi, R. Needham: “A Logic of Authentication”, ACM Transactions on Computer Systems, Vol. 8, No. 1, Feb 1990

[14] M. Bellare, D. Pointcheval, P. Rogaway: "Authenticated Key Exchange Secure Against Dictionary Attacks", Advances in Cryptology - Eurocrypt 2000 Proceedings, Lecture Notes in Computer Science Vol. 1807, B. Preneel ed, Springer-Verlag, 2000.

[15] M. Bellare, P. Rogaway: "Provably secure session key distribution: the three party case", Proceedings 27th Annual Symposium on the Theory of Computing, ACM, 1995.

[16] V. Shoup, A. Rubin: "Session-key distrubution using smart cards", in Proc. Eurocrypt '96, pp. 321-31, 1996

[17] J. C. Mitchell, V.Shmatikov, U. Stern: "Finite-State Analysis of SSL 3.0", Seventh USENIX Security Symposium, San Antonio, 1998, pages 201-216. Preliminary version presented at DIMACS Workshop on Design and Formal Verification of Security Protocols, September 1997

[18] V. Shoup: "On Formal Models for Secure Key Exchange", Theory of Cryptography Library Record 99-12, http://philby.ucsd.edu/cryptolib/ and invited talk at ACM Computer and Communications Security conference, 1999

[19] M. Wahl, T. Howes, S. Kille: "Lightweight Directory Access Protocol (v3)" RFC 2251, December 1997

[20] M. Wahl, A. Coulbeck, T. Howes, S. Kille: "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions" RFC 2252, December 1997

[21] M. Wahl, S. Kille, T. Howes: "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names" RFC 2253, December 1997

[22] T. Howes: "The String Representation of LDAP Search Filters" RFC 2254, December 1997

[23] T. Howes, M. Smith: "The LDAP URL Format" RFC 2255, December 1997

[24] M. Wahl: "A Summary of the X.500(96) User Schema for use with LDAPv3" RFC 2256, December 1997

[25] B. Kaliski: "PKCS #10: Certification Request Syntax Version 1.5" RFC 2314, March 1998

[26] S. Kent, R. Atkinson: “Security Architecture for the Internet Protocol” RFC 2401, November 1998

[27] S. Kent, R. Atkinson: “IP Authentication Header” RFC 2402, November 1998

[28] C. Madson, R. Glenn: “The Use of HMAC-MD5-96 within ESP and AH” RFC 2403, November 1998

[29] C. Madson, R. Glenn: “The Use of HMAC-SHA-1-96 within ESP and AH” RFC 2404, November 1998

[30] C. Madson, N. Doraswamy: “The ESP DES-CBC Cipher Algorithm With Explicit IV” RFC 2405, November 1998

[31] S. Kent, R. Atkinson: “IP Encapsulating Security Payload (ESP)” RFC 2406, November 1998

[32] D. Piper: “The Internet IP Security Domain of Interpretation for ISAKMP” RFC 2407, November 1998

[33] D. Maughan, M. Schertler, M. Schneider, J. Turner: “Internet Security Association and Key Management Protocol (ISAKMP)” RFC 2408, November 1998

[34] D. Harkins, D. Carrel: “The Internet Key Exchange (IKE)” RFC 2409, November 1998

[35] R. Glenn, S. Kent: “The NULL Encryption Algorithm and Its Use With IPsec” RFC 2410, November 1998

[36] R. Thayer, N. Doraswamy, R. Glenn: “IP Security Document Roadmap” RFC 2411, November 1998

[37] H. Orman: “The OAKLEY Key Determination Protocol” RFC 2412, November 1998

[38] R. Pereira, R. Adams: “The ESP CBC-Mode Cipher Algorithms” RFC 2451, November 1998

[39] C. Adams, S. Farrell: “Internet X.509 Public Key Infrastructure Certificate Management Protocols” RFC 2510, March 1999

[40] M. Myers, C. Adams, D. Solo, D. Kemp: “Internet X.509 Certificate Request Message Format” RFC 2511, March 1999

[41] S. Boeyen, T. Howes, P. Richard: “Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2” RFC 2559, April 1999

[42] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP” RFC 2560, June 1999

[43] R. Housley, P. Hoffman: “Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP” RFC 2585, May 1999

[44] S. Boeyen, T. Howes, P. Richard: “Internet X.509 Public Key Infrastructure LDAPv2 Schema” RFC 2587, June 1999

[45] R. Housley: “Cryptographic Message Syntax” RFC 2630, June 1999

[46] B. Ramsdell, Ed.: “S/MIME Version 3 Certificate Handling” RFC 2632, June 1999

[47] B. Ramsdell, Ed.: “S/MIME Version 3 Message Specification” RFC 2633, June 1999

[48] P. Hoffman, Ed.: “Enhanced Security Services for S/MIME” RFC 2634, June 1999

[49] R. Zuccherato: “Methods for Avoiding the ”Small-Subgroup” Attacks on the Diffie-Hellman Key Agreement Method for S/MIME” RFC 2785, March 2000

[50] M. Myers, X. Liu, J. Schaad, J. Weinstein: “Certificate Management Messages over CMS” RFC 2797, April 2000

[51] J. Reagle: “XML Signature Requirements” RFC 2807, July 2000

[52] M. Wahl, H. Alvestrand, J. Hodges, R. Morgan: “Authentication Methods for LDAP” RFC 2829, May 2000

[53] J. Hodges, R. Morgan, M. Wahl: “Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security” RFC 2830, May 2000

[54] P. Leach, C. Newman: “Using Digest Authentication as a SASL Mechanism” RFC 2831, May 2000

[55] H. Prafullchandra, J. Schaad: “Diffie-Hellman Proof-of-Possession Algorithms” RFC 2875, July 2000

[56] D. Eastlake 3rd, J. Reagle, D. Solo: "XML-Signature Syntax and Processing" RFC 3075, March 2001

[57] J. Boyer: "Canonical XML Version 1.0" RFC 3076, March 2001

[58] E. Rescorla: "Preventing the Million Message Attack on Cryptographic Message Syntax" RFC 3218, January 2002

[59] D. Eastlake 3rd, J. Reagle, D. Solo: "(Extensible Markup Language) XML-Signature Syntax and Processing" RFC 3275, March 2002

[60] R. Housley: "Cryptographic Message Syntax (CMS)" RFC 3369, August 2002

[61] R. Housley: "Cryptographic Message Syntax (CMS) Algorithms" RFC 3370, August 2002