

こういったコンピューターが実装されるのか、いつ実装されるのか、誰も正確に予想できません。一方で、NISTは、量子コンピューターでもやぶれない暗号文の開発プロジェクトを行っています(csrc.nist.gov/Projects/Post-Quantum-Cryptography)。

より一般的には、暗号アグリティーとは、さまざまなセキュリティ製品で使用されている特定のアルゴリズムを、これらの製品がサポートするビジネス・ワークフローに影響を与えることなく更新できる組織の能力を意味します(cryptosense.com/blog/achieving-crypto-agility)。

軽量暗号

また、現在の暗号アルゴリズムには、低消費電力デバイスでの利用という問題があります。NISTは、量子的な耐性があり、かつバッテリー駆動の機器でもCPUやメモリ資源を最小限に抑えたコンパクトな暗号スイートの開発を期待しています(csrc.nist.gov/projects/lightweight-cryptography)。

準同型暗号

準同型暗号方式は主に、プライバシーと要配慮データセットを共有する際に使用します。企業がプライベートデータを収集した場合、データの安全な保管と個人データの対象者のプライバシーの権利を尊重する責任があります。しかし多くの場合、企業は、第三者に分析を依頼することを希望します。このシナリオにおいて暗号化されていないデータの共有は重大なリスクになります。準同型暗号方式では、受け入れ会社は、データセット全体を暗号化した状態で、データ内のフィールドで統計計算を実行することができるため、この方式を使用すれば、このユースケースを解決することができます。例えば、顧客とのやり取りに関する分析を実行する場合、メールアドレスのようなアカウント識別子を解読することなく分析ツールはログオン回数を合計することができます。

ブロックチェーン

ブロックチェーンは、増加する取引レコードのリストを暗号技術で保護する概念です。各レコードはブロックと呼ばれ、ハッシュ関数にかけられます。1つ前に作成されたブロックのハッシュ値は、その次のブロックのハッシュの計算に追加されます。このようにして、連続する各ブロックはそれぞれ暗号的にリンクされています。各ブロックはチェーンの始点に到達するまで1つ前のブロックのハッシュ値を検証し、過去のトランザクションが変更されていないことを確認します。さらに、各ブロックには通常、1つ以上の取引のタイムスタンプと、取引自体に関わるデータが含まれています。

ブロックチェーンは公開台帳に記録されます。この台帳は、1台のコンピュータに個別のファイルとして存在するのではなく、分散型であることがブロックチェーンの最大の特徴です。单一障害点や侵害のリスクを軽減するため、台帳はピアツーピア(P2P)ネットワークで配布されます。そのため、ブロックチェーンの利用者は、お互いを同等に信頼することができます。同様に、ブロックチェーンのもう一つの特徴は、そのオープン性であり、誰もがブロックチェーン上のすべての取引を見ることができます。

ブロックチェーンは、さまざまな分野への応用が期待されています。例えば、金融取引やオンライン投票システム、アイデンティティ管理システム、公証、データストレージなどにブロックチェーンを活用すれば、データの完全性と透明性の確保に役立ちます。ただし、ブロックチェーンは、暗号通貨以外では比較的新しい技術のため、まだ応用例はそれほど多くありません。

ステガノグラフィー

ステガノグラフィー（文字通り「隠された文書」）はメッセージの存在を隠ぺいするための技術です。一般的に、情報は思われぬ場所に埋め込まれており、例えば、メッセージが絵の中に隠されています。コンテナ文書またはファイルはカバーテキストと呼ばれます。ステガノグラフィツールは、これを容易にする、あるいは逆に、カバーテキストの中に隠されたメッセージの存在を検出するために使用されるソフトウェアのことです。

ステガノグラフィは、情報を隠すために使われる場合、「隠ぺいによるセキュリティ」に該当しますが、これは通常、非推奨です。しかし、埋め込む前になんらかのメカニズムによってメッセージを暗号化することができるので、機密性を確保できます。この技術は、完全性や否認防止も提供できます。例えば、あるものが特定の時間に特定のデバイスで印刷されたことを示し、状況に応じてそれが本物であるか偽物であるかを証明することができます。

ステガノグラフィーの一例は、TCPパケットデータフィールド内でメッセージを暗号化し、隠しメッセージチャネルを作成することです。別のアプローチでは、画像ファイル内のピクセルの最下位ビットを変更することです。これにより、元の画像を大きく歪めることなく、有用な情報量を符号化することができます。同様の技術を、カーファイルとして、音声およびビデオファイルなど他のメディアタイプと共に使用することができます。

これらの方法は、コマンド&コントロールに使用されたり、データ損失防止(DLP)などの保護メカニズムを迂回してデータを密かに流出させるために使用される可能性があります。[\(blog.trendmicro.com/trendlabs-security-intelligence/steganography-and-malware-concealing-code-and-cc-traffic/\)](http://blog.trendmicro.com/trendlabs-security-intelligence/steganography-and-malware-concealing-code-and-cc-traffic/)。将来的な開発によって、ストリーミングメディアまたはVoIP (Voice over IP)でステガノグラフィーが使用されるようになるかもしれません。

レビュー アク ティビティ：

その他の暗号化技術の概要

次の質問にお答えください。

1. 分析会社と医療記録を共有する場合に最も有益な暗号化技術はなんですか？
2. あなたは、DLPソフトウェアの実施で顧客をサポートしています。検討中の2つの製品のうち、一方は画像データのステガナリシスをサポートしていますが、他方はサポートしていません。この機能を省略した場合どんなリスクがありますか？

レッスン5

概要

暗号の機能のタイプ（ハッシュアルゴリズム、対称暗号方式、非対称暗号方式）を概説し、これらがハイブリッド暗号化製品で機密性、完全性、認証、耐性を提供するためにどのように使用されているかを説明できる必要があります。また、これらの限界や弱点と、暗号化攻撃の一般的なタイプを特定できる必要があります。最後に、量子、ブロックチェーン、準同型暗号方式、およびステガノグラフィーなどその他のコンセプトを概説できる必要があります。

レッスン6

公開鍵インフラストラクチャを実装する

レッスン概要

デジタル証明書と公開鍵インフラストラクチャ (PKI)は、ほとんどのプライベートネットワークやパブリックネットワークで識別、認証、データの機密性を管理するために使用される重要なサービスです。発行可能な証明書の種類を把握し、こうしたシステムを構成およびサポートする際に有効な管理原則を適用できることが重要になります。

レッスンの目的

このレッスンの内容は、以下のとおりです。

- 証明書と認証局を実装する。
- PKI管理を実装する。

トピック6A

証明書と認証局を実装する



対象試験範囲

3.9 与えられたシナリオに基づいて、公開鍵インフラストラクチャを実装することができる

デジタル証明書は認証局(CA)によって検証される、身元の公的な証明です。証明書は、身元の証明のほか、Webサーバー通信の保護やメッセージの署名など、さまざまな目的で発行されます。証明書の発行は、セキュリティ管理者としての日常業務の重大な一部になる可能性があります。

公開鍵と秘密鍵の使用

公開鍵暗号方式により、他者と安全に通信する場合や、他者に送信するメッセージを認証する場合に、暗号化の鍵を配信する問題が解決されます。

- 他者から機密のメッセージを送信してもらう場合、自身の公開鍵を提供し、メッセージを暗号化してもらうことができます。メッセージはその後、自分自身しか知らない秘密鍵でのみ復号できます。
- 他者に対して、自分自身であることを証明する場合、シグネチャを作成し、秘密鍵でそのシグネチャを暗号化することで署名できます。相手にはシグネチャを復号できるように公開鍵を渡します。秘密鍵を知っているのは本人のみであるため、誰もがあなただけが署名を作成できたと確信できます。

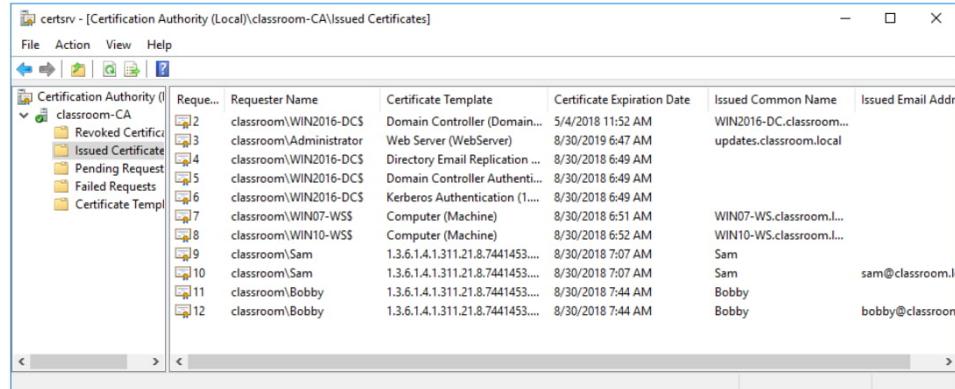
公開鍵暗号方式の基本的な問題は、通信している人物について本当は知らない可能性があることです。またこのシステムは、中間者攻撃に対して脆弱です。この問題はeコマースで特に明らかです。ショッピングサイトや銀行業務が、本当に本物が管理していることを、どのようにして確認できるでしょうか？サイトが通信を保護するために公開鍵を配布しているという事実は、実際の身元の証明にはなりません。公開鍵を使用しているサイトと直接通信していると、どうして分かるのでしょうか？また、正規のサーバーからの送信だと思っているものは中間者によって傍受や改ざんがなされていないと、どのように確認できるでしょうか？

公開鍵インフラストラクチャ(PKI)では、公開鍵の所有者が本人であることを証明することを目標にしています。PKIでは、公開鍵を発行する人はデジタル証明書を取得する必要があります。その証明書の有効性は、認証局(CA)によって保証されます。CAの有効性は、さまざまなモデルを使用して確立できます。

認証局

認証局(CA)は、証明書の発行と保証に責任を担うエンティティです。プライベートCAは、社内通信向けに組織内で設定できます。Windows Serverを含む、ほとんどのネットワークオペレーティングシステムに証明書サービスがあります。ですが、パブリックまたはB2Bの通信の場合、各当事者がCAを信頼する必要があります。サードパーティのCAサービスには、IdenTrust、Digicert、Sectigo/Comodo、GoDaddy、GlobalSignなどがあります。CAの機能は次のとおりです。

- CAがサービスを提供するユーザーのコミュニティにとって役立つさまざまな証明書サービスを提供する。
- 証明書の有効性と申請者の身元を確認する（登録）。
- ユーザー、政府、規制当局、金融機関などの企業によるCAへの信頼を確立する。
- 証明書を保存・管理するサーバー（リポジトリ）を管理する。
- 鍵と証明書のライフサイクル管理を実行し、特に無効な証明書を取り消す。



The screenshot shows the 'certsrv - [Certification Authority (Local)\classroom-CA\Issued Certificates]' window. The left pane displays a tree view with 'classroom-CA' expanded, showing 'Revoked Certificates', 'Issued Certificate', 'Pending Requests', 'Failed Requests', and 'Certificate Template'. The right pane lists 12 issued certificates with the following details:

Requester Name	Certificate Template	Certificate Expiration Date	Issued Common Name	Issued Email Address
classroom\WIN2016-DC\$	Domain Controller (Domain...)	5/4/2018 11:52 AM	WIN2016-DC.classroom...	
classroom\Administrator	Web Server (WebServer)	8/30/2019 6:47 AM	updates.classroom.local	
classroom\WIN2016-DC\$	Directory Email Replication ...	8/30/2018 6:49 AM		
classroom\WIN2016-DC\$	Domain Controller Authenti...	8/30/2018 6:49 AM		
classroom\WIN2016-DC\$	Kerberos Authentication (1...	8/30/2018 6:49 AM		
classroom\WIN07-WSS	Computer (Machine)	8/30/2018 6:51 AM	WIN07-WS.classroom.l...	
classroom\WIN10-WSS	Computer (Machine)	8/30/2018 6:52 AM	WIN10-WS.classroom.l...	
classroom\Sam	1.3.6.1.4.1.311.21.8.7441453...	8/30/2018 7:07 AM	Sam	
classroom\Sam	1.3.6.1.4.1.311.21.8.7441453...	8/30/2018 7:07 AM	Sam	sam@classroom.local
classroom\Bobby	1.3.6.1.4.1.311.21.8.7441453...	8/30/2018 7:44 AM	Bobby	
classroom\Bobby	1.3.6.1.4.1.311.21.8.7441453...	8/30/2018 7:44 AM	Bobby	bobby@classroom.local

Microsoft Windowsサーバー CA。（スクリーンショットはMicrosoftからの許可を得て使用。）

PKI信頼モデル

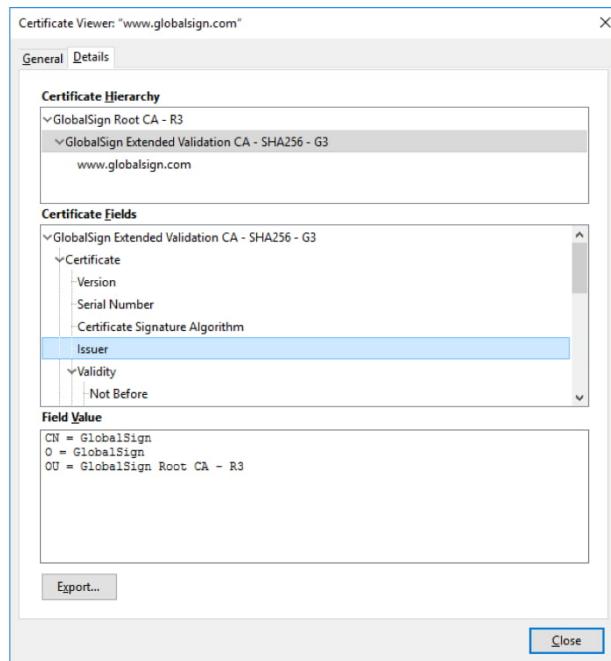
信頼モデルは重要なPKIコンセプトで、ユーザーやさまざまなCAがどのようにお互いに信頼できるかを示します。

単一CA

このシンプルなモデルでは、単一のCAがユーザーに証明書を発行します。ユーザーはこのCAによって発行された証明書のみ信頼します。このアプローチの問題は、単一CAサーバーが大々的に公開されていることです。もし侵害されると、PKI全体が崩壊します。

階層型（中間CA）

階層型モデルの場合、単一CA（ルートと呼ばれる）は、いくつかの中間CAに証明書を発行します。中間CAでは対象（リーフまたはエンドエンティティ）に証明書を発行します。このモデルには、さまざまな中間CAにさまざまな証明書ポリシーを設定できるという利点があり、ユーザーは特定の証明書が何のために設計されているかを明確に認識できます。各リーフ証明書は、証明書パスに沿ってルートCAに遡ることができます。これは、**証明書チェーン**または信頼の連鎖とも呼ばれます。ルートの証明書は自己署名されています。階層型モデルの場合、ルートは依然として単一障害点になります。ルートが損傷または侵害されている場合、構造全体が崩壊します。ですがこれを軽減するために、ルートサーバーをオフラインにできます。これは、通常のCAアクティビティのほとんどが中間CAサーバーによって処理されるためです。



証明書のパス。リーフ証明書(www.globalsign.com)が中間拡張認証CAによって発行され、そのCAの証明書はルートCAによって発行されています。
(スクリーンショットはMicrosoftからの許可を得て使用。)

もう1つの問題は、相互認証、つまり別の組織のCAを信頼する機会が限られることです。2つの組織がルートCAを共有することに同意することができますが、より多くの組織が参加するにつれて増大する運用上の問題につながる可能性があります。実際にはほとんどのクライアントで複数のルートCAが信頼されるように構成されています。

オンラインCAとオフラインCA

オンラインCAは、認証署名要求の受け入れと処理、認証失効リストの公開、他の認証管理タスクの実行に使用できるCAです。ルートCAを侵害することによりリスクが高くなることから、セキュアな構成ではルートを**オフラインCA**にする必要があります。これはネットワークから切断され、通常は電源がオフの状態に保たれることを意味します。中間CAを追加したり、更新するにはルートCAをオンラインにする必要があります。

登録局とCSR

登録とは、エンドユーザーがCAでアカウントを作成し、証明書を要求するために認証されるプロセスです。ユーザーが認証され、身元が保証される正確なプロセスは、CAの実装によって決定されます。例えば、Windows Active Directoryネットワークでは、多くの場合、ユーザーとデバイスはActive Directoryに認証させるだけで、CAに自動登録できます。商用CAでは、さまざまなテストを実行し、対象が本人であることを確認できます。正当なユーザーにのみ証明書を発行することを保証することはCAの利益になりますが、そうでなければ、その評判は損なわれます。

! プライベートネットワーク (Windowsドメインなど) の場合、異なる種類の証明書を発行する権利は慎重に管理される必要があります。Windows CAでは、各証明書の種類のアクセス許可がサポートされるため、どのアカウントで証明書を発行できるかを選択できます。

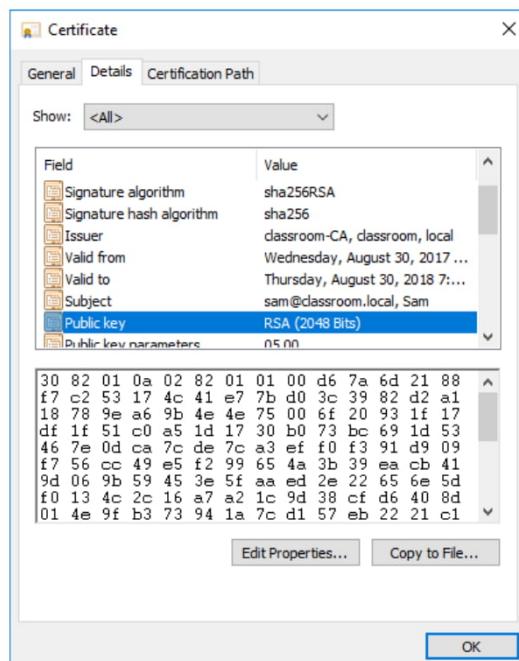
対象が証明書を取得する際は、**証明書署名要求(CSR)**を作成し、CAに送信します。CSRは、Base64 ASCIIファイルで、対象が証明書で使用する、公開鍵などの情報が含まれます。

CAでは、証明書をレビューし、情報が有効であることを確認します。Webサーバーの場合、これは単に対象の名前と完全修飾ドメイン名(FQDN)が同じであることを確認し、ドメインのWHOISレコードで識別された、そのドメインの管理責任者によってCSRが開始されたことを確認することを意味します。要求が承諾されると、CAでは証明書に署名をし、対象に送信します。

登録機能は、CAによって1つまたは複数の**登録局(RA)**に委任されることがあります。こうしたエンティティでは、身元確認を行い、エンドユーザーの代わりにCSRを送信しますが、実際に証明書に署名したり発行したりしません。

デジタル証明書

デジタル証明書は、基本的には対象の公開鍵のラッパーです。これには公開鍵のほか、対象と証明書発行者や保証人に関する情報が含まれます。証明書はデジタル署名され、特定のCAによって対象に発行されたことを証明します。対象は、人間のユーザー（メッセージの署名を許可する証明書など）やコンピューターサーバー（機密トランザクションをホストするWebサーバーなど）になります。



デジタル証明書の詳細。（スクリーンショットはMicrosoftからの許可を得て使用。）

デジタル証明書は、国際電気通信連合が承認し、Internet Engineering Taskforce（インターネット技術特別調査委員会）によって標準化されたX.509標準に基づきます(tools.ietf.org/html/rfc5280)。公開鍵インフラストラクチャ(PKI)ワーキンググループがこうした標準の開発を管理します。また、RSAでは、公開鍵インフラストラクチャの使用を促す**公開鍵暗号化規格(PKCS)**と呼ばれる一連の規格も作成しています。

証明書の属性

X.509標準では、証明書に含まれる必要のあるフィールドや属性を定義します。主要なフィールドの一部を次の表で紹介します。

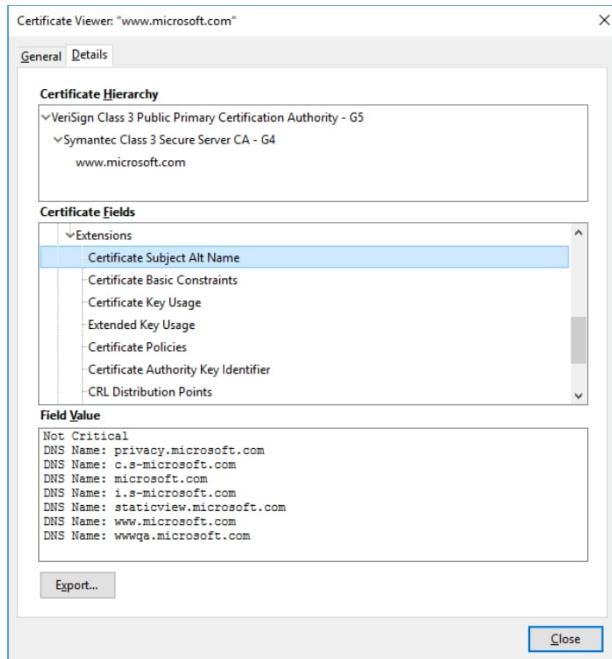
フィールド	用途・用例
シリアル番号	CAのドメイン内で証明書を個別に識別する番号。
署名アルゴリズム	証明書の署名にCAによって使用されるアルゴリズム。
発行者	CAの名前。
有効期限 (いつから/いつまで)	証明書が有効な期間の日付と時刻。
対象	識別名(DN)で表現される証明書保持者の名前。 この中で、共通名(CN)部分は通常、サーバーの完全修飾ドメイン名(FQDN)かユーザーの電子メールアドレスのいずれかに一致するはずです。
公開鍵	証明書保持者によって使用される公開鍵とアルゴリズム。
拡張	V3証明書は、分かりやすい対象名や発行者名、連絡先電子メールアドレス、意図される鍵の使用など、拡張属性で定義できます。
サブジェクトの別名(SAN)	この拡張フィールドは、DNS名またはホストが識別される名前を特定する推奨メカニズムです。

対象名属性

証明書が最初に導入されたときには、**共通名(CN)**属性は、www.comptia.orgなどサーバーがアクセスされるFQDNを特定するために使用されていました。ですがこの使用法は、計画的ではなくカスタマイズによって増えました。CN属性には、さまざまな種類の情報を含めることができるために、ブラウザで正しく解釈するのは困難になります。その結果、CN属性は、対象の身元確認方法として廃止されました(tools.ietf.org/html/rfc2818#section-3.1)。

サブジェクトの別名(SAN)拡張フィールドは、ドメイン名を含むさまざまな種類の識別子を表現するために構成されています。証明書がSANで構成される場合、ブラウザではそれを確認し、CN値は無視されるはずです。しかし、すべてのブラウザと実装が最新の標準に準拠しているわけではないので、FQDNをCNにする方が依然として安全です。

またSANフィールドでは、www.comptia.orgとmembers.comptia.orgなど、証明書がさまざまなサブドメインを表すようにさせることもできます。



異なるサブドメイン用にサブジェクトの別名で構成されたMicrosoftのWebサイト証明書。
(スクリーンショットはMicrosoftからの許可を得て使用。)

特定のサブドメインをリスト化する方が安全ですが、新しいサブドメインが追加される場合は、新しい証明書が発行される必要があります。*.comptia.orgなどのワイルドカードドメインは、親ドメインに発行された証明書がすべてのサブドメインで有効（単一レベルまで）になります。

Field	Value
Public key parameters	05 00
Authority Key Identifier	KeyID=0f80611c823161d52f2...
Subject Key Identifier	a50d532930871c2818ad0c65f...
Subject Alternative Name	DNS Name=*.comptia.org, DN...
Enhanced Key Usage	Server Authentication (1.3.6....)
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Certificate Policies	[1]Certificate Policy:Policy Ide...
Authority Information Access	[1]Authority Info Access: Acc...

DNS Name=*.comptia.org
DNS Name=comptia.org

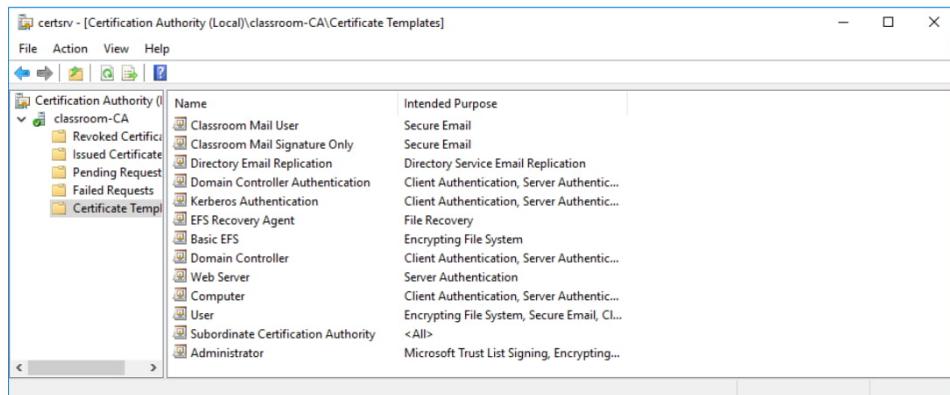
ワイルドカードドメインで構成されたCompTIAのWebサイト証明書。これにより`https://comptia.org`または`https://www.comptia.org`のいずれからアクセスできます。（スクリーンショットはMicrosoftからの許可を得て使用。）

証明書の種類

証明書ポリシーでは、CAによって発行される証明書タイプのさまざまな使用法を定義します。これらは、標準の証明書テンプレートとして構成できます。

証明書のタイプは、鍵使用法(KU)属性を構成することで設定されます。拡張鍵使用法(EKU)属性 (MicrosoftではEnhanced Key Usageと呼ばれる)は、使用法を定義する補完的な方法です。使用される一般的な値には、サーバー認証、クライアント認証、コード署名、または電子メール保護が含まれます。EKU属性は、KU属性よりも柔軟ですが、標準以外の定義またはベンダー専用の定義を使用すると問題が発生する可能性があります。

拡張機能はクリティカルとしてタグ付けできます。つまり証明書を処理するアプリケーションは拡張機能を正しく解釈できる必要があり、それ以外の場合、証明書は拒否されます。KU拡張機能がクリティカルとマークされている場合、アプリケーションでKU値を解決できない場合、証明書は拒否されるはずです。例えば、これにより、Webサーバーに送信されるトラフィックを暗号化するために発行された証明書が、電子メールメッセージの署名に使用されないようにします。



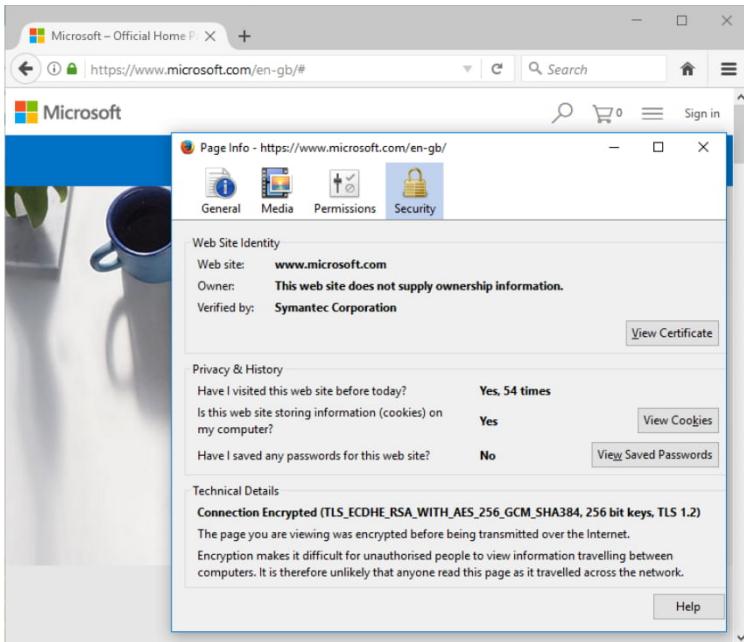
Windows Server CAの証明書テンプレート。(スクリーンショットはMicrosoftからの許可を得て使用。)

Webサーバー証明書のタイプ

サーバー証明書は、eコマースサイトなど、ユーザーが機密を保持すべきデータを送信するサイトの身元を保証します。公開鍵暗号方式と信頼モデルの問題の1つは、誰もがPKIソリューションを設定できることです。また、my-bank-server.fooなど信ぴょう性があるように見えるドメイン名を登録するのも簡単です（実際のドメインはmybank.foo）。単に証明書を持っていればサイトが信頼できるという甘い考え方で証明書を信用すると、詐欺にあう可能性があります。また評判の悪いサイトが、サードパーティのCAから証明書を取得し、ブラウザによって自動的に信頼することで、金融機関としての身元を検証することになっている場合もあります。

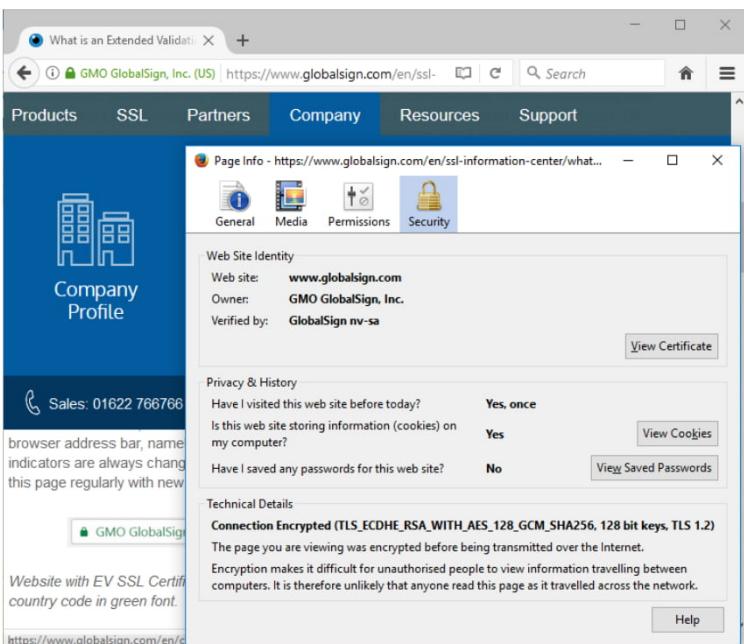
異なるグレードの証明書を使用して、セキュリティのレベルを実現できます。例えば、オンラインバンクでは、マーケティングデータを収集するサイトよりも高いセキュリティが必要になります。

- ドメイン認証(DV) — 特定ドメインの所有権を証明します。これは、承認されたドメインの連絡先への電子メールに応答するか、テキストレコードをドメインに公開することで証明できます。このプロセスは、侵害に対して非常に脆弱である可能性があります。



ドメイン認証証明書。南京錠のみが表示され、ブラウザでは所有者が検証されていないことが報告されています。(スクリーンショットはMicrosoftからの許可を得て使用。)

- 拡張認証(EV) – 署名対象の法的な身元と、ドメインやソフトウェアの管理について、より厳密なチェックを必要とするプロセスの対象となります。EV標準は、CA/ブラウザフォーラム(cabforum.org)で管理されます。EV証明書は、ワイルドカードドメインには発行できません。



GlobalSignからの拡張認証証明書。検証済み所有者が南京錠の横に緑で表示されます。
(スクリーンショットはGlobalSign, Inc.の許可を得て使用。)

他の証明書の種類

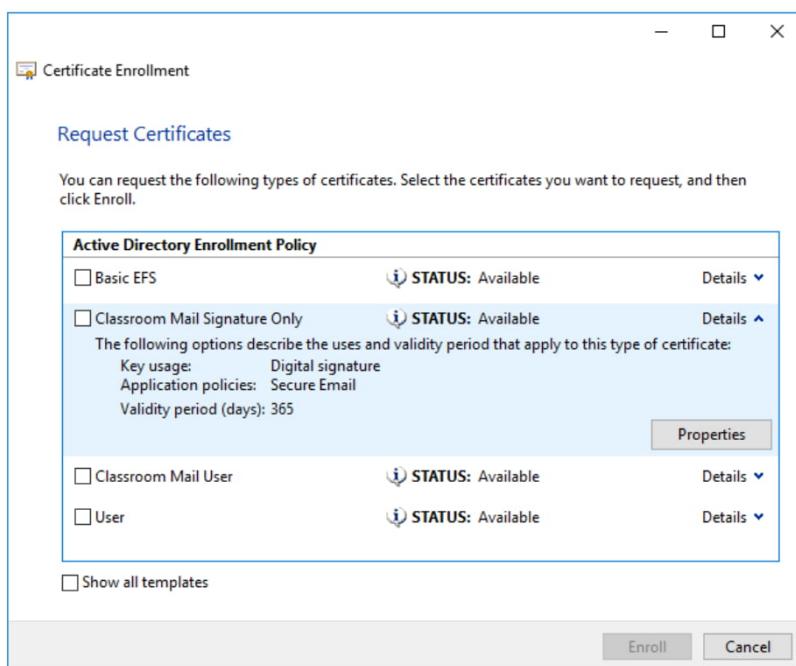
Webサーバーだけが身元の検証が必要なシステムではありません。その他にもさまざまな目的のために多くの証明書の種類があります。

マシン/コンピューターの証明書

機能に関わらず、マシン（サーバー、PC、スマートフォン、タブレット）に証明書を発行する必要がある場合があります。例えば、Active Directoryドメインの場合、マシン証明書はドメインコントローラー、メンバーサーバー、またはクライアントのワークステーションに対して発行できます。有効なドメイン発行証明書のないマシンでは、ネットワークのリソースにアクセスできない可能性があります。マシン証明書は、ルーター、スイッチ、ファイアウォールなどのネットワークアプライアンスに対して発行できます。SAN属性と、多くの場合はCN属性も、マシンのFQDN（ホスト名とローカルドメイン部分）に設定されるべきです。

電子メール/ユーザー証明書

電子メール証明書は電子メールメッセージの署名と暗号化に使用できます。これには通常 Secure Multipart Internet Message Extensions (S/MIME)またはPretty Good Privacy (PGP) が使用されます。ユーザーの電子メールアドレスは、SAN属性やCN属性として入力される必要があります。Windows Active Directoryなどのディレクトリベースのローカルネットワークの場合、さまざまなユーザー証明書の種類が必要になる可能性があります。例えばADでは、標準ユーザー、管理者、スマートカードログオン/ユーザー、**回復エージェント**ユーザー、Exchangeメールユーザー向けのユーザー証明書テンプレートがあります（署名と暗号化には個別のテンプレートを使用）。各証明書テンプレートにはさまざまな鍵使用法の定義があります。



証明書を要求。CAでは、さまざまな鍵の使用方法の指定（ファイルの暗号化、電子メールの署名、電子メールの暗号化など）により、いくつかのユーザータイプの証明書テンプレートを用意しています。
(スクリーンショットはMicrosoftからの許可を得て使用。)

コード署名証明書

コード署名証明書は、CAによるある種の身元確認と検証プロセスを経て、ソフトウェア発行者に発行されます。そして、発行者は、ソフトウェアアプリケーションやブラウザプラグインの正当性を保証するために、実行可能ファイルやDLLに署名します。PowerShellなどの一部のスクリプト環境でも、有効なデジタル署名が必要になります。CN属性には、FQDNではなく、“CompTIA Development Services, LLC”のような組織名が設定されます。

ルート証明書

ルート証明書は、CAそのものを識別する証明書です。ルート証明書は自己署名済みです。通常、ルート証明書は最低2048ビットの鍵長を使用します。多くのプロバイダーは4096ビットに切り替えています。ルート証明書のCN属性は、FQDNではなく「CompTIA Root CA」などの組織/CA名が設定されます。

自己署名入り証明書

マシン、Webサーバーまたはプログラムコードは、**自己署名入り証明書**でデプロイできます。自己署名入り証明書は、オペレーティングシステムまたはブラウザによって信用できないとマークされますが、管理者はこれを無視できます。

レビュー アク ティビティ： 証明書と認証局

次の質問にお答えください。

1. 階層信頼モデルの主な弱点は何ですか。
2. 対象はどのようにしてCAから証明書を取得しますか。
3. どの暗号化情報がデジタル証明書に保存されますか。
4. 証明書拡張属性がクリティカルとしてマークされるとはどのような意味ですか。
5. あなたはセキュアなWebアプリケーションを開発しています。あなたがプログラムのパブリッシャーであることを示すためには、どの種類の証明書を要求すべきですか。
6. 複数の特定のサブドメインラベルでサーバーの識別をサポートするために、Webサーバー証明書で使用される拡張フィールドは何ですか。

トピック6B

PKI管理を実装する



対象試験範囲

3.9与えられたシナリオに基づいて、公開鍵インフラストラクチャを実装することができる
4.1与えられたシナリオに基づいて、適切なツールを利用して組織のセキュリティを評価するこ
とができる（OpenSSLのみ）

あなたはセキュリティの専門家として、プライベートネットワークに公開鍵インフラストラクチャ(PKI)証明書サービスをインストールし、管理する必要がある可能性がかなりあります。また、パブリックPKIプロバイダーから証明書を取得し、管理する必要があります。このトピックでは、PKIのインストールと構成、証明書のトラブルシューティングと取り消しについて説明します。

証明書および鍵管理

鍵管理は、鍵のライフサイクルのさまざまな段階における運用上の考慮事項を指します。鍵のライフサイクルには次の段階が関与します。

- 鍵の生成 – 選択した暗号アルゴリズムを使用して要求される強度のセキュアな鍵ペアを作成します。
- 証明書の生成 – 鍵ペアの公開部分が対象（ユーザまたはコンピュータ）のものであることを識別するために、適切な鍵の使用法でデジタル証明書としてCAにより署名を求めて提出します。この時点で、証明書を要求する対象の身元を確認し、対象が身元確認に合格する場合のみに証明書が発行されるようにすることが重要です。
- 保管 – ユーザーは秘密鍵を安全に保管するための手順を講じて、不正なアクセスや不正使用が防止されるようにする必要があります。また、秘密鍵は紛失したり、損傷したりしないようにすることが重要です。
- 取り消し – 秘密鍵が侵害される場合、鍵ペアは取り消して、ユーザーが公開鍵を信用しないようにできます。
- 有効期限と更新 – 取り消されていない鍵ペアは、一定期間が過ぎると無効になります。鍵や証明書に「寿命」を与えることでセキュリティを強化できます。証明書は新しい鍵のマテリアル（材料）で更新できます。

鍵管理は、1名の管理者または機関がプロセスを管理する一元型、または各ユーザーが自身の鍵に責任を持つ分散型で管理できます。

証明書および鍵管理は、適切に管理しないと重大な脆弱性を示す可能性があります。脅威アクターが秘密鍵を取得すると、データの機密性と、識別/認証システムの両方が危険にさらされることになります。脅威アクターが有効に見える署名済み証明書を作成できたとすると、対応するユーザー アカウントとコンピュータ アカウントが自動的に信頼されるため、ネットワークから大量の情報を簡単に収集できるようになります。最後に、暗号化に使用された鍵が誤って破壊される場合、バックアップまたは鍵回復メカニズムがない限り、その鍵を使用して暗号化されたデータにアクセスできなくなります。

鍵の回復とエスクロー

ルートCAの秘密鍵などの鍵は、可能な限り最高の技術的および手続き的なアクセス制御の対象となる必要があります。そのような鍵が侵害されると、数百数千におよぶシステムによって処理されるデータの機密性と完全性が危険にさらされることになります。そのような重要な暗号化鍵へのアクセスは、記録および監査される必要があり、通常は**M-of-N制御**の対象となります。つまり、システムへのアクセスが許可された管理者N名のうち、アクセスが許可されるにはM名がその場にいる必要があります。Mは1より大きく、NはMより大きい数にする必要があります。例えばM = 2でN = 4の場合、4名の管理者のうち2名がその場にいる必要があります。鍵管理を実施する権限のある従業員は慎重に精査される必要があり、こうした従業員が会社を辞める場合は、十分な注意を払う必要があります。



*M-of-N制御のもう1つの使用方法は、複数のストレージデバイス間で鍵を分割することです
(3つのUSBスティックで、完全な鍵の再作成にはそのうちの2本が必要になるなど)。*

データの復号に使用される鍵が紛失または損傷した場合、鍵のバックアップを作成している場合を除き、暗号化データは回復できません。鍵の保管における重要な問題は、鍵のバックアップを複数作成すると、その鍵が侵害されないようにすることが急激に難しくなることです。ですが鍵のバックアップをしない場合、保管システムは単一障害点になります。鍵回復では、鍵のバックアップや紛失した鍵で暗号化されたデータを回復するためのセキュアな処理を定義します。このプロセスでは、アーカイブされた鍵への不正アクセス（と不正使用）を防止するため**M-of-N制御**を使用する場合があります。**エスクロー**とは、何かが独立して保持されていることを意味します。鍵管理において、これはサードパーティでの鍵のアーカイブを指します。これは、独自に安全に鍵を保管できない組織にとって有用なソリューションですが、そのサードパーティに多大な信頼を寄せなければなりません。

証明書の有効期限

証明書は、証明書タイプごとに設定されたCAポリシーに従って限られた期間で発行されます。ルート証明書の有効期限は長い（10年以上）可能性がありますが、Webサーバーとユーザー証明書は1年間のみである可能性があります。通常、証明書は有効期限が切れる前に更新されます。ユーザーが有効な証明書を所有している場合、新しい証明書を要求する場合よりも、（身元確認に関して）必要な手続きが少なくなります。証明書を更新する場合、既存の鍵を使用する（特に鍵更新と呼ばれる）か、新しい鍵を生成する（証明書の鍵を再生成する）ことができます。古い鍵が十分な長さであると見なされなくなった場合や、鍵の侵害が懸念された場合、新しい鍵が生成されることがあります。

証明書の有効期限が切れると、それに関連する鍵ペアをどう処理するかという問題が発生します。鍵は、アーカイブするか破棄できます。鍵を破棄する方がより安全ですが、その鍵を使用して暗号化されたデータが読み取れなくなるという欠点があります。鍵をアーカイブするか、破棄するかは、鍵がどのように使用されたかによって大きく異なります。ソフトウェアの場合、鍵はデータを上書きすることで破棄できます（データを削除するだけでは安全ではありません）。ハードウェアに保管される鍵は、指定の消去手順に従って破棄するか、デバイスを破壊して破棄できます。

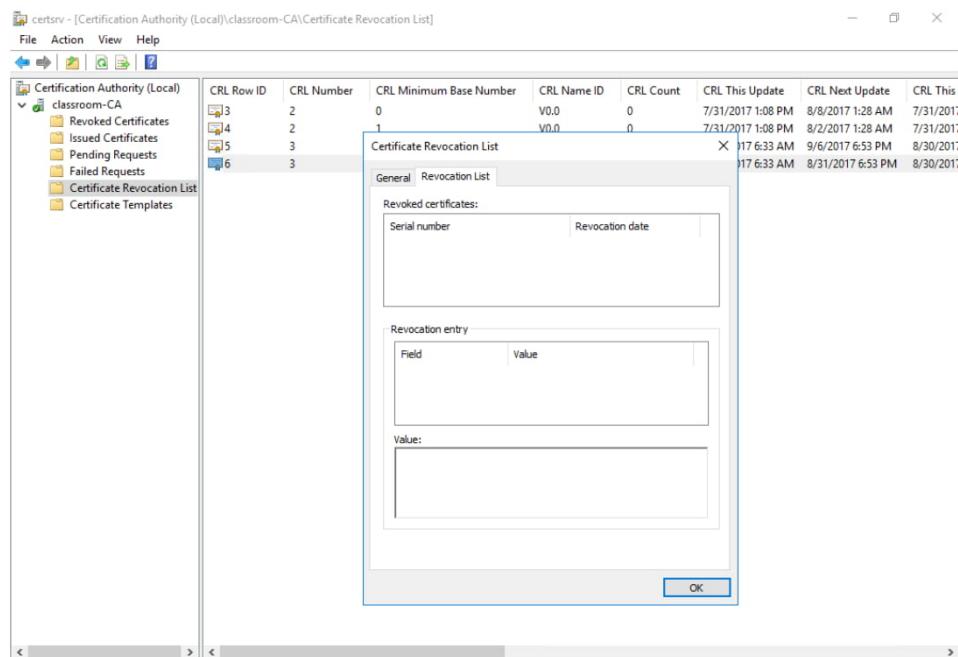
証明書失効リスト

証明書は取り消すか一時停止にできます。

- 取り消された証明書は無効になり、「取り消しの取り消し」をしたり、復元することはできません。
- 一時停止した証明書は再び有効にできます。

証明書は、さまざまな理由で所有者またはCAによって取り消すか一時停止にできます。例えば、証明書またはその秘密鍵が侵害された場合、ビジネスが閉鎖された場合、ユーザーが会社を辞めた場合、ドメイン名が変更された場合、証明書が何らかの方法で悪用された場合などがあります。このような理由は、「不明」、「鍵の侵害」、「CAの侵害」、「置き換え」、「運用の停止」などの選択肢でコード化されます。一時停止された鍵は、コード「証明書保留」が与えられます。

従って、証明書が有効であるか、取り消されているか、または一時停止されているかをユーザーに通知するための何らかのメカニズムが必要になります。CAでは、取り消された証明書や一時停止された証明書すべての**証明書失効リスト(CRL)**を維持する必要があります。これは階層全体で配布できます。



Windows Certificate Servicesによって公開されたCRL — 最新のCRLには取り消された証明書が1つ含まれます。(スクリーンショットはMicrosoftからの許可を得て使用。)

CRLシステムでは、最新のCRLが公開されていないため、証明書が取り消されてもクライアントに受け入れられる可能性があるというリスクがあります。さらなる問題は、ブラウザ（または他のアプリケーション）がCRLチェックを実行するように構成されていない可能性があることですが、現在は、レガシーブラウザソフトウェアだけの傾向があります。

オンライン証明書状態プロトコルレスポンダー

最新情報を提供するまた別の方法は、**オンライン証明書状態プロトコル(OCSP)**サーバーで証明書の状態を確認することです。このサーバーはOCSPレスポンダーとも呼ばれます。これは、CRL全体を返すのではなく、要求された証明書の状態を通知するだけです。OCSPレスポンダーサービスの詳細は、証明書で公開されるはずです。



ほとんどのOCSPサーバーでは、証明書データベースを直接クエリして、証明書のリアルタイムの状態を取得できます。他のOCSPサーバーは実際にはCRLに依存し、CRL発行間隔によって制限されます。

OCSPの問題の1つは、要求に応答するジョブがリソースを大量に消費し、OCSPレスポンダーを実行する発行CAに高い要求を課す可能性があることです。またOCSPレスポンダーはクライアントのブラウザ要求を監視し、記録するために使用できるため、プライバシーの問題があります。OCSPステーピングでは、SSL/TLS WebサーバーでタイムスタンプされたOCSP応答をCAから定期的に取得することで、こうした問題を解決します。クライアントがOCSPリクエストを送信すると、Webサーバーはタイムスタンプ付きのレスポンスを返し、クライアントがOCSPレスポンダーに自らコンタクトする必要はありません。

証明書のピン留め

証明書が、SSL/TLSなどのトランSPORTプロトコルによって使用される場合、クライアントやサーバー、または中間CAとルートCAが証明書を提供した対象との間の信頼の連鎖が侵害される可能性があります。脅威アクターは、悪意があるが信頼された証明書を（一種のプロキシまたは中間者攻撃を使用して）信頼の連鎖に置き換えることができる場合、安全とされる接続をスヌーピングできる可能性があります。

ピン留めとは、クライアントがサーバーやコード署名されたアプリケーションによって示される証明書を検査する際に、適切な証明書を検査していることを確認するためのいくつかの手法を指します。これは、アプリケーションコードに証明書のデータを組み込むか、*HTTP Public Key Pinning (HPKP)*と呼ばれるHTTPヘッダー経由で1つまたは複数の公開鍵をHTTPブラウザに送信することで実現できます。

HPKPには重大な脆弱性があるため、非推奨になりました(developer.mozilla.org/en-US/docs/Web/HTTP/Public_Key_Pinning)。代わりとなるメカニズムは*Certificate Transparency Framework*になります。

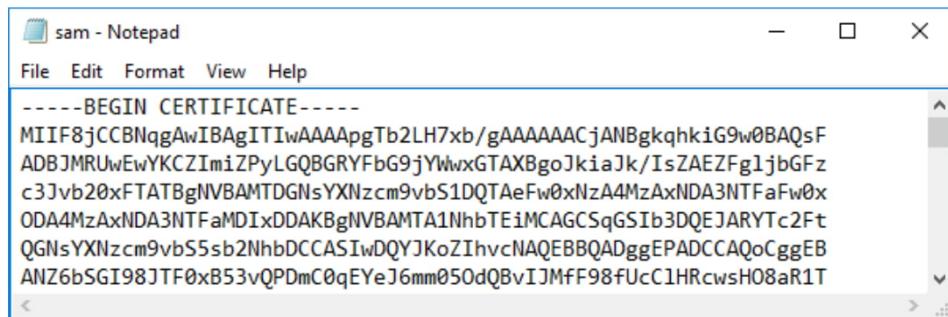
証明書フォーマット

さまざまなシステム間で交換するためのデジタルファイルとして、証明書をエンコードするさまざまなフォーマットがあります。

エンコード

暗号データ（証明書と鍵の両方）は**識別符号化規則(DER)**を使用してバイナリとして処理されますが、バイナリ形式ファイルは一般的に使用されていません。

より一般的には、バイナリデータはBase64**プライバシー強化メール(PEM)**エンコードを使用する**ASCII**テキスト文字として表現されます。ASCII形式のデータは「BEGIN CERTIFICATE」文字列などの記述的なヘッダーが使用されています。



```
-----BEGIN CERTIFICATE-----
MIIF8jCCBNqgAwIBAgITIwAAAApgTb2LH7xb/gAAAAAACjANBgkqhkiG9w0BAQsF
ADBjMRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxGTAXBgoJkiaJk/IzAEZFg1jbGFz
c3Jvb20xFTATBgnVBAMTDGnsYXNzcm9vbS1DQTaeFw0xNzA4MzAxNDA3NTFaFw0x
ODA4MzAxNDA3NTFaMDIxDDAKBgNVBAMTA1NhbTEiMCAGCSqGSIb3DQEJARYTc2Ft
QGNsYXNzcm9vbS5sb2NhDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
ANZ6bSGI98JTF0xB53vQPdmC0qEYeJ6mm050dQBvIJMFF98fUcC1HRcwsH08aR1T
```

メモ帳で開かれたBase64でエンコードされた.CERファイル。
(スクリーンショットはMicrosoftからの許可を得て使用。)