賢いことの結果を予測することは不可能です。  
  
– LARS KNUDSEN  
  
パスワードは、より一般的な概念であるセキュリティプロトコルの一例にすぎません。  
プリンシパルが信頼関係を確立するために使用する手順を指定します。  
チャレンジレスポンス認証やKerberosなど、すでにいくつかのプロトコルに遭遇しています。  
典型的なセキュリティシステムは、人、会社、電話、コンピューター、カードリーダーなどの多くのプリンシパルで構成され、ファイバー、Wi-Fi、セルラーネットワーク、Bluetooth、赤外線などのさまざまなチャネルを使用して通信し、物理デバイスでデータを伝送します。銀行カードや交通チケットなど。  
これらは、電話でうそをつく人々、敵対的な政府の電波妨害、または偽造者が電車のチケットのデータを変更するなどの悪意のある行為にシステムが耐えられるように設計されています。  
たとえば、ユーザーにマシンにパスワードを入力してログオンさせる場合、ユーザーはそれを正しいマシンに入力できると暗黙的に想定しています。  
したがって、プロトコルの評価には2つの質問があります。1つ目は、脅威モデルが現実的かどうかです。  
 プロトコルは、リーダーを介してバッジをスワイプして建物に入るなど、非常に単純な場合があります。  
たとえば、レストランで高級ワインのボトルを注文する場合、標準のプロトコルでは、ワインウェイターがメニューを提供します（価格は表示されますが、ゲストは表示されません）。  
これは、プライバシーを提供するように進化しました（ゲストは価格を知りません）  
 そして否認防止（ワインがオフだったと後で不平を言うことはできません）  
Matt Blazeは、チケット検査、航空セキュリティ、投票からの他の非技術的なプロトコルの例を[260]で提供しています。  
技術的には、プロトコルはより複雑になり、常に改善されるとは限りません。  
しかし、キーレスエントリへの移行により、悪意のある人がキーを実際よりも車に近づけるようにリレーデバイスを構築する方法を見つけたため、車の犯罪が再び増加しています。  
ヨーロッパは2000年代後半にこの動きをしましたが、アメリカは2010年代後半に追いついています。  
また、盗難者がPINを知らなくても盗まれたチップカードを店で使用できるようにするプロトコルの失敗があり、銀行が修正するのに数年かかりました。  
  
マシンに対して人間を認証するために使用される主要なメカニズムとして、パスワードとPINは依然として多くのコンピュータセキュリティの基盤です。  
リモートキー入力は、開始するのに適しています。  
それらを殺した攻撃は「グラバー」でした。これはコードを記録し、後でそれを再生するデバイスです。  
1ガレージのドアがあると、それはさらに悪化します。  
ただし、実装者はしばしば、トライステート入力を理解し、代わりにそれらをバイナリとして処理するために十分に注意深くデータシートを読まずに、212を取得します。  
また、チップには再試行ロックアウトロジックがないため、攻撃者は4.3を循環できます。  
 –簡単な認証最初の対策は、ロックとロック解除に別々のコードを使用することでした。  
次に、16ビットのパスワードは短すぎます。  
そして、1990年代半ばまでに、すべての可能なコードを次々と試すことができるデバイスが登場しました。  
範囲内に100台の車両がある駐車場で泥棒が運転すると、車がライトを点滅させるのに役立ち、1分足らずで報われます。  
メーカーは「40億を超えるコード」を誇らしげに宣伝しました。  
各車にはまだ1つまたは2つのコードしかなく、グラバーはまだうまく機能していました。  
車の場合、これはすべてのディーラースタッフ、そしておそらく国の自動車登録機関を意味するかもしれません。  
そして銀行は誰かを現金で買うために送るのは好きではありません。  
たとえば、地元のスイミングプールの月間シーズンチケットには、単にバーコードがあります。  
ただし、オンラインの場合、静的なパスワードは危険です。 Miraiボットネットは、パスワードを変更できないwifi接続のCCTVカメラを採用することで成功しました。  
これにより、暗号化認証プロトコルが実現します。  
4.3誰がそこに行くのですか？ –単純な認証  
トークンにはボタンが1つあります。押すと、最初にシリアル番号を送信してから、同じシリアル番号とそれに続く乱数で構成される認証ブロックを送信します。すべて、デバイス固有のキーを使用して暗号化され、ガレージバリアに送信されます（通常は434MHzの無線による。ただし、赤外線も使用される）データを暗号化する方法の説明を次の章に延期し、キーKで暗号化されたメッセージXに対して{X} Kを書き込むだけです。  
この本の第2版でこれらの問題に触れてから12年後、チップは回収されていません。  
4.3。  
 –簡単な認証これは標準のプロトコル表記であるため、ゆっくりと説明します。  
ブレース内のすべてが暗号化され、暗号化によってTとNがバインドされ、それらの値が不明瞭になります。  
検証は簡単です。ガレージはTを読み取り、対応するキーKTを取得し、メッセージの残りの部分を解読し、ナンスNが以前に見られていないことを確認し、最後に平文にTが含まれていることを確認します。  
 右側は名前（つまり、一意のデバイス番号）を意味します  
もう1つは、プロトコルへの攻撃について説明し始めると、あるプリンシパル宛てのメッセージが別のプリンシパルに傍受されて再生されたことがわかる場合があります。  
nonceは、メッセージの鮮度を保証するものであれば何でもかまいません。  
それらの間には微妙な違いがあります。たとえば、さまざまな種類のリプレイアタックに対する抵抗のレベルや、システムのコストと複雑さを増加させる方法などです。  
このようなデバイスのキー管理は非常に簡単です。  
これは、アクセストークンを実装する一般的な方法であり、スマートカードでも広く使用されています。  
システムを完全に破壊し、システムのユーザーのふりをすることを可能にするマスターキーを抽出するために、攻撃者はガレージの中央サーバーを侵害する必要があります（これにより、このキーが改ざんから保護される可能性があります）。耐性のあるスマートカードまたはハードウェアセキュリティモジュール）  
しかし、まだエラーの余地があります。  
これは暗号化によってマスクできます。  
どちらの場合も、可能なキーは216のみであり、4.3であっても十分ではありません。  
 –単純な認証それらはランダムに見える2。  
例は、前払いユーティリティメーターの世界から来ています。  
南アフリカで広く使用されていた初期のメーターの1つは、ナンスが前回と異なることのみを確認しました。  
有効と見なされた[93]。  
乱数を使用する場合、ロックは過去のコードの多くを記憶する必要があります。  
さらに、誰かが車を借りて、十分なロック解除コードを記録し、後でレンタルロットに戻ってそれを盗む可能性があります。  
カウンターを選択した場合、問題は同期です。  
。  
1つの一般的な製品は16ビットのカウンターを使用し、解読されたカウンター値が16を超えてインクリメントされた最後の有効なコードでない場合にアクセスを許可します。  
、有効なコードが入力されてからカウンター値が17から32,767回インクリメントされた場合、2回目のプレスでロックが開きます（カウンターがロールオーバーして、0が65,535の後継になります）。  
これは多くのアプリケーションで細かいですが、6つの適切に選択されたアクセスコード（たとえば、値0、1、20,000、20,001、40,000、40,001）を取得できる泥棒は、システムを完全に破壊する可能性があります。  
 したがって、単純なトークン認証メカニズムを設計することも、見た目ほど簡単ではありません。また、製品が低グレードの敵のみを引き付けると想定している場合、この仮定は時間の経過とともに失敗する可能性があります。  
多くのプリンタ会社は、認証メカニズムをプリンタに組み込んで、本物のトナーカートリッジが使用されるようにしています。  
科学的な楽器からゲーム機まで、他のあらゆる種類の業界がこの動きに加わっています。  
しかし、裁判官は、Lexmark対SCCの場合、ベンダーは顧客をロックするために見つけることができる最高の暗号化技術者を雇う権利を有していたが、競合他社も2詳細について検討する権利を有していると判断した。セクション5.3.1.2では、確率論における誕生日の定理について説明します。  
誰が行くの？  
これは深刻な軍備競争を引き起こしました。それは後の章で随時会います。  
セキュリティメカニズムは多くのビジネスモデルをサポートするために使用され、通常、デバイスの所有者を悪意のある人から保護するのではなく、デバイスの所有者がやりたいことをやめさせます。  
  
1995年以降、ヨーロッパで販売されているすべての車には「暗号化対応のイモビライザー」が必要で、2010年までにほとんどの車にリモートコントロールドアのロック解除がありましたが、ほとんどの車にはフォールバックメタルキーが付いているため、車に乗ることができますキーフォブのバッテリーが平らであっても。車のキーがステアリングロックに挿入されると、エンジンコントローラは、短距離無線を使用して、ランダムなnビット数からなるチャレンジをキーに送信します。  
周波数が低い（125kHz）  
エンジンコントローラにE、車のキーのトランスポンダにT、トランスポンダとエンジンコントローラ間で共有される暗号化キーにK、ランダムチャレンジにNを書くと、プロトコルは次のようになります。E ffi！ T：N T ffi！ E：T、{T、N} Kこれは理論的には健全ですが、セキュリティメカニズムの実装は、多くの場合、最初の2、3回は失敗します。  
攻撃には、プロトコルエラー、ピアキー管理、脆弱な暗号、輸出管理法で義務付けられている短いキーの組み合わせが含まれていました。  
Stephen Bonoらは2005年に、40ビットのキーを持つブロック暗号を使用していることを発見しました。これは、たった2つの応答からのブルートフォースで計算できます[297]。  
2010年から、フォード、トヨタ、ヒュンダイは後継製品であるDST80を採用した。  
誰が行くの？  
 [2048]。  
さらに悪いことに、車の種類によっては、プロトコルのバグもあり、キーの多様化で排他的論理和が使用されています。KT= T ffi KM。  
また2007年に、誰かがPhilips Hitag 2暗号を公開しました。これにも48ビットの秘密鍵がありました。  
2016年までに、攻撃はラップトップ上で8回の認証試行と1分の計算を要しました。彼らは日産、三菱そしてシボレーと一緒にすべてのフランスとイタリアのメーカーからの自動車に対して働いた[748]。  
メガモス暗号を含み、バーミンガムとナイメーヘンの研究者（Roel Verdult、Flavio Garcia、Barı¸sEge）がそれを解読した[1952]によって、リバースエンジニアリングされた車の鍵製造ツールが2008年に登場しました。  
フォルクスワーゲンは、彼らの営業秘密が侵害されたと主張して、Usenix 2013での彼らの仕事の提示を停止するようロンドンの高等法院で差し止め命令を受け取りました。  
2年間の議論の後、訴訟はいずれの側にも責任を認めることなく和解した。  
攻撃者が96ビットキーの各16ビットワードを次々に書き換えて、一度に16ビットのキーを検索できるため、プロトコル攻撃もあります。これにより、攻撃に必要な時間が数日から数分に短縮されます[1953]。  
多くのフォルクスワーゲンの実装では、車やトランスポンダ間でキーを分散させることはしませんでしたが、一度に数百万台の車に固定されたグローバルマスターキーを使用しました。  
車が脆弱かどうかは簡単にわかります。スペアキーを購入してみてください。  
前の所有者が購入した妻の2005年型レクサス用のスペアキーがあります。  
もう1つの例は、衛星TVセットトップボックス内のサブスクライバースマートカードです。これについては、後で説明します。  
 攻撃。  
4.3。  
 – 1週間で簡単な認証。  
まったく同じことがここでも当てはまります！しかし、最悪の事態はまだ続きました：パッシブキーレスエントリーシステム（PKES）  
チャレンジレスポンスが良さそうだったため、自動車メーカーは、金属製のキーではなく、ダッシュボードのプッシュボタンだけで自動車を始動しました。  
マーケティングのピッチでは、キーをポケットやハンドバッグに入れておく限り、心配する必要はありませんでした。車は近づくとロックが解除され、離れるとロックされ、コントロールに触れると自動的に始動します。  
 まあ、ボタンを押して車のロックを解除する必要はなくなりました。信号を増幅または中継するデバイスを泥棒が簡単に使用できるようになりました。  
ホールのテーブルに鍵を置いたままにすると、車のドアが開いて離れます。  
そして、何年にもわたる車の盗難の後、統計は2017年に急増し、英国では56％多くの車両が盗まれ、2018年にはさらに9％が盗まれました[823] 4。  
それ以来、政治家やマーケティング担当者が邪魔をするようになりました。  
次に、イモビライザーベンダーのマーケティング担当者は、独自のアルゴリズムで自動車会社をロックしたいと考えました。  
 まあ、少なくとも2つの自動車メーカーが加速度計をキーフォブに入れているので、キーが動いていないと機能しません。  
地元の警察は、昔ながらの金属製のステアリングホイールロックを使用するようにアドバイスしています。私たちの自治会は鍵をビスケット缶に保管することを推奨しています。それで我々はその車を取り戻し、適切な押しボタン式のリモートロック付きの中古車を手に入れました。  
ただし、暗号自体はリレー攻撃を修正できません。適切な修正は、ウルトラワイドバンド（UWB）に基づく新しい無線プロトコルです。  
これは適切に行うにはかなり複雑であり、新しい802.15.4z拡張インパルス無線の設計はSrdjan Capkunと同僚[1764]によって記述されています。 4公平を期すために、これはリレー攻撃のみによるものではありませんでした。高額の盗難の約半分は、車の盗難キットをグローブボックスの下のオンボード診断ポートに接続することを含むようです。  
これらのプロトコルの更新には、莫大な産業投資があるため、何年もかかります。  
誰が行くの？  
このようなチップは、BluetoothプロトコルとNFCプロトコルの両方を置き換える可能性がありますが、すべて互換性があるとは限りません。低レートパルス（LRP）がある  
 一部が独占的なバリアント。  
ロックは、チャレンジ/レスポンスプロトコルの唯一のアプリケーションではありません。  
応答は、ナンスのハッシュ、パスワード、および要求されたURI [715]で構成されます。  
たとえば、Voice-Over-IP（VOIP）のプロトコルであるSIPでクライアントとサーバーを認証するために使用されます。  
パスワードを平文で送信するよりもはるかに優れていますが、キーレスエントリのように、中間者攻撃に苦しんでいます（受益者は偽物です）  
  
チャレンジ/レスポンスの最も明白な使用法は、おそらく2要素認証です。  
彼らは小さな電卓のように見えるかもしれません（そしていくつかはそのように機能します）  
ログインするときは、7桁のランダムなナンスが表示されます。  
デバイスは、企業のセキュリティサーバーと共有する秘密鍵を使用してこれらの11桁を暗号化し、結果の最初の7桁を表示します。  
このプロトコルを図4.1に示します。  
正式には、Sはサーバー、Pはパスワードジェネレータ、PINはユーザーの個人識別番号、Uはユーザー、Nはナンスです。Sffi！ U：N U ffi！ P：N、PIN P ffi！ U：{N、PIN} K U ffi！ S：{N、PIN} Kこれらのデバイスは1980年代初頭に登場し、最初は電話会社に、次に1990年代には銀行でスタッフが使用するようになりました。  
そして彼らは働きます。米国国防総省は2007年に、DoD Common Access Cardに基づく認証システムにより、昨年のネットワーク侵入を46％削減したと発表しました[320]。  
私の銀行の1つは、ログオンごとに新しいコードを生成する小さな計算機を提供し、チャレンジの代わりに口座番号の最後の4桁を使用して新しい受取人を認証することもできます。  
、4.3。  
 –単純な認証N？  
。  
。  
 ん？  
しかし、これはまだ完全なものではありません。  
また、「一度多くの銀行がワンタイムパスワードを使用すると、フィッシャーマンはスクリプトを書き換えてリアルタイムの中間者攻撃を行う」と述べ、これも広まった。  
  
チャレンジ/レスポンス認証プロトコルが最初に使用されたのはおそらく「軍人か敵か」（IFF）で軍隊にあった  
1930年代と1940年代の戦闘機の速度の増加により、ジェットエンジン、レーダー、ロケットの発明により、防空部隊が敵の航空機と自分の航空機を区別することがさらに困難になりました。  
これらは第二次世界大戦で最初に発見され、レーダーに照らされた飛行機が識別番号をブロードキャストして友好的な意図を伝えることができるようになりました。  
今日、典型的な防空システムはレーダー信号でランダムなチャレンジを送信し、友好的な航空機は正しい応答で自分自身を識別できます。  
問題の1つは、南アフリカ空軍（SAAF）の将校から聞いた次のストーリーに示されています  
この本の初版で出版された後、ストーリー4.3。  
 –簡単な認証が争われました–以下で説明します。  
「MIG-in-the-middle」のストーリーはその後、民間伝承の一部となり、チャレンジ/レスポンスプロトコルでリアルタイムに攻撃を実行する方法をうまく示しています。  
彼らの目標は、ナミビアを白人の支配下に置き、クライアント政府（UNITA）を課すことでした。  
南アフリカ国防軍は主に小さな白人集団の徴集兵で構成されていたため、死傷者を制限することが重要でした。そのため、ほとんどの南アフリカ兵士はナミビアに留まり、北への戦闘はUNITA軍によって行われました。ん？  
 ANGOLA NAMIBIA SAAF MIG N？  
彼らの空軍優位が失われたというこの証拠は、プレトリア政府がナミビアを4.3に引き渡すことを決定するのを助けました。  
 –簡単な認証反乱–それ自体が数年後に南アフリカで多数決に至る道への大きな一歩です。  
数年後、SAAFの職員がキューバ人がどのようにそれを止めたのか教えてくれました。  
その後、MIGは鋭く向きを変え、SAAFの防空壕を進んで進み、IFFに挑戦しました。  
私の情報提供者によると、これはプレトリアの一般職員に衝撃を与えました。  
この物語が私の本の最初の版で出版された後、私は物語の詳細に異議を唱えたSA Communications Security Agencyの元役員から連絡を受けました。  
したがって、彼は電子工作はもっと原始的なものであったに違いないと述べた。  
いずれにせよ、この物語は中間者攻撃の別の例を示しています。  
また、パスワード計算機に対しても機能します。フィッシングサイトは、マークをログオンに招待し、同時に彼の銀行とのログオンセッションを開きます。  
中間者の攻撃を阻止することは見かけよりも難しく、複数の防御層が必要になる場合があります。  
アカウントに新しい受取人を追加するなどのリスクの高いトランザクションの場合、どちらの銀行も、受取人の口座番号で認証コードを計算するように要求します。  
支払いに2分と数十桁の入力が必要な場合、多くの顧客は数字を間違え、あきらめ、コールセンターに電話するか、迷惑をかけて他の場所で銀行に預けます。  
これらすべてについては、銀行業務と簿記の章でさらに説明します。  
彼らはゲームにも適用されます。  
誰が行くの？  
4.3.4反射攻撃  
対空砲手が友好的な航空機を攻撃するのを防ぐように設計されたチャレンジレスポンスIFFシステムが戦闘機にも配備されなければならなかったとします。  
しかし今、戦闘機が敵爆撃機に挑戦するとき、爆撃機はその挑戦を戦闘機の翼手に反射し、正しい応答を得て、それを独自の応答として送り返すことができます：F ffi！ B：N B ffi！ F 0：N F 0 ffi！ B：{N} K B ffi！ F：{N} K取引所に2つの当事者の名前を含めるなど、これを阻止する方法はいくつかあります。  
これは、認証の根底にある信頼の前提の微妙さを示すのに役立ちます。  
しかし、あなたが知っているのはそれだけです。  
キーKのすべてのコピーが適切に動作すると信頼されている可能性のある機器に安全に保持されているとさらに想定し、{B、N} Kが表示された場合、コールサインBの航空機が2000マイル。  
今では、IFFのすべてのプロトコル設計の側面を理解していると思うかもしれません。  
レーダーがパッシブであるため、リターンは弱く、IFFがアクティブであるため、IFFトランスミッターからの信号は通常、同じ航空機のレーダーリターンよりもはるかに広い範囲で聞こえます。  
自分のメッセージを覚えたり認識したりできる必要があります。エニグマのメッセージは、ドイツ人がIFFに問い合わせることにより、通常のレーダー範囲の2倍でイギリスとアメリカの爆撃機を計画していたことを明らかにしました。  
たとえば、NATOモードXIIには32ビットの暗号化チャレンジがあり、問い合わせ信号ごとに異なる有効なチャレンジが生成されます。通常、1秒あたり250です。  
モードXIIでは、方向性と飛行時間を使用してリレーが困難になります。  
第23章でIFFに戻ります。  
1種類の攻撃から保護できるプロトコル（自分の側で撃墜される）  
 助けにならないかもしれません。  
  
参加者の認証に使用される情報を反映または偽装する中間者攻撃が数多く見られます。  
私たちがすでに見た1つの例は、最後に見たチケットのみを記憶する前払いメーターです。そのため、2つのチケットAとBのコードを次々とコピーすることで、無制限に再チャージできます。ABABAB...  
プロペラシャフトが回転すると、センサーがパルスを送信します。これにより、メーターはタクシーがどこまで行ったかを把握できます。  
規制で許可されている以上の速度で運転したいトラック運転手は、同様のデバイスを使用して一部のパルスを破棄できるため、運転速度が遅いかまったくないようです。監視システムだけでなく、制御システムもメッセージ操作攻撃に対して強化する必要があることがよくあります。  
特定のアプリケーションに関する後の章で、メッセージ操作を伴うプロトコル攻撃の多くの例を見ます。  
4.5環境の変化  
良い例は、ATM詐欺の世界から来ています。  
結局、銀行は犠牲者の多くがユトレヒトの近くの特定の充填所で自分の銀行カードを使用したことに気づきました。  
彼がカードリーダーからそれを制御するPCまでの回線を盗んだことが判明しました。彼のタップはカードから磁気ストライプの詳細を記録し、その間彼は眼球を使ってPINをキャプチャしました[54]。  
私たちの地元の充填ステーションで、200人以上の顧客が自分のカードがタイのATMで使用されていたことに突然気付きました。  
 まあ、磁気ストライプカードとPINを管理するための標準が1980年代初頭にIBMやVISAなどの組織によって開発されたとき、エンジニアは2つの仮定をしていました。  
（使用された類推は、磁気ストリップがあなたの名前であり、PINがパスワードであるというものでした。）  
したがって、PINパッドからサーバーに向かう途中で、PINを暗号化することだけが「明らかに」必要でした。磁気ストリップのデータは、カードリーダーからクリアテキストで送信できます。  
カード偽造の蔓延により、主に1980年代後半に極東で、銀行は磁気ストリップに認証コードを導入するようになりました。  
これら2つの環境変化の組み合わせにより、元のシステムアーキテクチャの背後にある仮定が破壊されました。  
これらの変化は非常にゆっくりと、そして非常に長い期間にわたって行われていたため、業界は問題の発生を認識していませんでした。  
4.6選択されたプロトコル攻撃  
シンガポールはミリタリーIDを兼ねた銀行カードで実験を行った。  
 一部の企業は、多種多様なカードやキーを持ち歩く必要をなくすために、幅広いトランザクションで使用できる多機能認証デバイスを推進しています。  
しかし、これも見た目ほど簡単ではないかもしれません。  
では、マフィアは銀行取引の認証を攻撃するプロトコルをどのように設計するのでしょうか？  
ポルノウェブサイトにアクセスする人々が「年齢の証明」を求められるのはよくあることでした。これには通常、サイト自体または年齢確認サービスのいずれに対しても、クレジットカード番号の提供が含まれます。  
図4.3に示すように、ポルノサイトは「中間者攻撃」を仕掛ける可能性があります。  
 ディーラーから、コインディーラーの顧客の役割を果たす。  
貧しい人はそれを承諾し、マフィアは金貨を受け取り、何千人もの人々が月末に彼らのカードへの莫大な請求について突然不満を言うとき、ポルノサイトは金と一緒に消えました[1032]。  
それ以来、多くのプロトコルは、それ自体は安全ですが、ユーザーが他のアプリケーションで同じキーを再利用するように仕向けられると、破られる可能性があることが示されています[1032]。  
これは、アクセス制御に関する次の章の主題になります。  
 複数のアプリケーションでは危険ですが、自分のアプリケーションのセキュリティを他の人がブートストラップできるようにすることは、まったくばかげている可能性があります。  
セクション3.4.1で説明したように、悪意のある人は、SIMスワップ詐欺によってそのシステムを攻撃することを学びました。電話会社を標的にして、電話を紛失したと主張し、交換用のSIMカードを入手します。  
4.7暗号化キーの管理  
非常に重要な認証プロトコルのクラスがもう1つあります。暗号化キーの管理に使用されるプロトコルです。  
4.7.1復活したアヒルの子  
車両は初期の例を提供しました。  
同じ問題が、運転手が運転時間と速度を監視するためにトラックが使用する装置であるタコグラフでも逆に起こりました。  
しかし、キーはどのように管理できますか？  
これについては、セクション14.3で詳しく説明します。  
デフォルトの「動作する」モードでは、新しいHomeplugデバイスは最初に表示されるキーを信頼します。新しいWi-Fiエクステンダーが隣のWi-Fiとかみ合う場合は、リセットボタンを押して、もう一度試してください。  
TOFUアプローチは、フランクスタジャノと私がタコグラフ作業のコンテキストで行った分析の結果、「復活したアヒルの子」としても知られています。電子デバイスでそのような誤ったインプリンティングが発生した場合、それを殺して新生児状態に復活させる方法が必要です–リセットボタンがそれを行います[1819]。  
4.7.2リモートキー管理  
基本的な技術は、分散型コンピュータシステムのキーを管理するために1970年代後半から開発され、ATMが初期のアプリケーションです。  
鍵配布プロトコルの背後にある基本的な考え方は、2人のプリンシパルが通信したい場合、信頼できるサードパーティを使用してそれらを紹介する可能性があるということです。  
したがって、通信する2つのプリンシパルを「アリス」と「ボブ」、信頼できるサードパーティを「サム」と呼びます。  
（たとえば、自動車ディーラーが交換用キーを自動車と嵌合させるプロトコルでは、アリスが自動車、ボブがキー、サムが自動車メーカーである場合があります。）  
1。  
2。  
それぞれにキーのコピーが含まれています。最初のキーは暗号化されてアリスだけが読み取れ、2番目のキーは暗号化されてボブだけが読み取れます。  
次にアリスはボブに電話し、2番目の証明書を紹介として提示します。  
これで、アリスはキーを使用して暗号化されたメッセージをボブに送信し、代わりにボブからメッセージを受信できます。  
証明書の有効期限が切れない場合は、特権が取り消されたユーザーを扱う際に深刻な問題が発生する可能性があります。  
サムは、アリスの名前、ボブの名前、使用するセッションキー、およびタイムスタンプで構成されるメッセージを作成します。  
彼は両方の暗号文をアリスに渡します。  
彼女は、このセッションキーを使用して暗号化し、送信したいすべてのメッセージを彼に送信します。  
4.7.3 Needham-Schroederプロトコル  
既存の鍵配布プロトコルの多くは、1978年に登場したNeedham-Schroederプロトコルから派生しています[1426]。  
次のように実行されます。メッセージ1 A！ S：A、B、NAメッセージ2 S！ A：{NA、B、KAB、{KAB、A} KBS} KASメッセージ3 A！ B：{KAB、A} KBSメッセージ4 B！ A：{NB} KABメッセージ5 A！ B：{NB ffi 1} KABここでアリスがイニシアチブを取り、サムに「私はアリスです。ボブと話したいのですが、ランダムなナンスはNAです。」サムは彼女にセッションキーを提供します。彼女が彼と共有する鍵。  
彼はまた、この鍵をボブに伝えるための証明書を彼女に与えます。  
このプロトコルには微妙な問題があります。ボブは、サムから（アリス経由で）キーKABを受け取ると想定する必要があります。  
これは必ずしもそうではありません。アリスはステップ2と3の間に1年待っていた可能性があります。  
しかし、対戦相手（たとえばチャーリー）がアリスのキーを手にしたことがあれば、それを使用して他の多くのプリンシパルとのセッションキーをセットアップできます。  
つまり、失効が問題になります。サムはこれまでに行ったすべてのことを完全にログに記録する必要があり、これらのログは、将来のある一定の時間にプリンシパルの名前が期限切れにならない限り、サイズが永久に大きくなります。  
単純化した見方では、ニーダムとシュローダーはちょうどそれを誤解したということです。スーザンパンチョとディーターゴルマン（私が同情している）が主張した見解  
1978年は、優しく穏やかな世界でした。その後、コンピュータセキュリティは「悪者」を排除することを懸念していましたが、今日では、「敵」がシステムのユーザーの中にいると予想しています。  
これらの仮定の下では、プロトコルは健全なままです。  
4.7.4 Kerberos  
特にマシンがローカルエリアネットワークを介してリソースを共有する場合、WindowsとLinuxの両方の認証の基本的なメカニズムの一部になりました。  
これにより、スケーラブルなアクセス管理が可能になります。  
最初に、アリスはパスワードを使用して認証サーバーにログオンします。  
彼女が正しいパスワードを持っていると仮定すると、彼女はKASを制御し、チケット認可サーバーSによって制御されるリソースBにアクセスするために、次のプロトコルが実行されます。  
これが許可される場合、適切なキーKABを含むチケット{TS、L、KAB、A} KBSが作成され、使用するためにアリスに与えられます。  
彼女は、タイムスタンプTAをリソースに送信してチケットを検証します。これにより、1ずつ増加したタイムスタンプを送り返すことにより、チケットが有効であることを確認します（これは、チケットを正しく復号化してキーKABを抽出できたことを示します）。  
Needham-Schroederプロトコルの失効問題は、ランダムなノンスではなくタイムスタンプを導入することで修正されました。  
これで新しい脆弱性が発生しました。つまり、さまざまなクライアントとサーバーのクロックが同期しなくなる可能性があります。より複雑な攻撃の一部として、意図的に非同期化されることもあります。そのSのプロトコルは信頼されています。警察が令状を持って出廷した場合、彼らはサムに鍵を渡してトラフィックを読み取らせることができます。  
サードパーティへの信頼がまったくないか、ほとんどないプロトコルは、一般に公開鍵暗号方式を使用します。これについては、次の章で説明します。  
たとえば、Googleを使用してDoodleにログインし、DoodleにGoogleカレンダーの更新を許可すると、DoodleのWebサイトからGoogleにリダイレクトされ、ログインできるようになります（または以前のログインのマスターCookieに依存します）。  
Doodleがカレンダーサービスのアクセストークンを提供します[863]。  
OAuthはユーザー認証用に設計されておらず、アクセストークンはクライアントに強くバインドされていません。  
上記の例では、GoogleでのDoodleへのアクセスを取り消すことができるようにしたいので、Doodleはバックグラウンドで有効期間の短いアクセストークンのみを取得します。  
OpenID Connectは、GoogleまたはFacebookアカウントを使用して新聞にログインするときに使用するものです。  
4.7.5実用的な鍵管理  
しかし、何万人ものスタッフのための暗号化されたパスワードと、多数のデバイスのための鍵もあるかもしれません。  
どのように管理されますか？  
座って、いくつのキーが必要か、どのように生成するか、どれくらいの期間サービスを維持する必要があるか、最終的にどのように破棄するかについて考える必要があります。  
そして、アプリケーションが進化するにつれて、物事はうまくいきません。来年の機能をサポートするためのヘッドルームを提供することが重要です。  
しかし、標準的な方法もありません。  
銀行では、通常、ハードウェアセキュリティモジュールと呼ばれる専用の暗号化プロセッサを使用します。これについては、後で詳しく説明します。  
  
上で見たような微妙な困難、および保護特性が誤解されているかもしれない微妙な仮定に依存する多くの方法により、研究者たちは正式な方法をプロトコルに適用するようになりました。  
このプロセスは、特定のプロトコルの根底にある仮定を明確にするのに役立つことがよくあります。  
最もよく知られているものの1つは、発明者のバロウズ、アバディ、ニーダム[357]にちなんで名付けられた信念の論理、またはBAN論理です。  
他の研究者は、CSPなどの主流の正式な手法やIsabelleなどの検証ツールを適用しています。  
これはドイツでは「ゲルトカルト」としてまだ使用されていますが、他では使用されなくなっています（南アフリカではNet1、ベルギーではプロトン、フランスではモネオ、そしてCOPACと呼ばれるVISA製品でした）。  
したがって、私はこのエディションから悲惨な詳細を削除することにしました。第2版​​はオンラインで無料なので、詳細をダウンロードして読むことができます。  
しかし、それらにも限界があります。  
たとえば、Larry Paulsonは、1998年にIsabelleの定理証明者を使用してSSL / TLSプロトコルを検証しました。それ以来、毎年約1つのセキュリティバグが発見されています。  
この場合、正式な方法の失敗はありませんでした。それは単に、見ている必要がない場所を攻撃者に伝えただけです。  
構造化プログラミング技術がソフトウェアを系統的に設計し、重要なものを除外しないことを保証することを目的とするのと同じように、堅牢なプロトコル設計は主に明示性についてです。  
 メッセージに明示的に記載する必要があります。  
カウンター、タイムスタンプ、ランダムチャレンジによって提供される鮮度、および暗号化の使用方法に関する他の問題があります。  
数学的暗号解読に基づくものから、電力消費やタイミングなどのサイドチャネルを悪用するものまで、攻撃全体を行うことが難しくなるため、プロトコルのランダム性は、食欲を刺激するために、他のレイヤーの堅牢性を助けることがよくあります。マイクロプローブまたはレーザーを含む物理的な攻撃。  
4.9まとめ  
プロトコルは、プリンシパルがシステムで信頼関係を確立するために使用する手順を指定します。たとえば、IDに対する要求の認証、資格情報の所有権の証明、リソースに対する要求の確立などです。  
セキュリティプロトコルは、リモートの車のドアロックから軍事IFFシステムを介して、分散型コンピュータシステムの認証に至るまで、あらゆる種類のシステムで使用されています。  
ミドルパーソン攻撃、修正攻撃、反射攻撃、リプレイ攻撃など、多くの問題に悩まされる可能性があります。数学的手法を使用してプロトコルの正しさを検証することは役立ちますが、すべてのバグをキャッチできるわけではありません。  
その結果、攻撃は依然として私たちが長年使用してきたプロトコルで頻繁に見られ、場合によってはセキュリティの証拠があると考えていたプロトコルでさえ見つかることもあります。  
独自のプロトコルを設計しないでください。専門家に助けを求め、研究コミュニティによる徹底的な査読のために設計が公開されていることを確認してください。  
。  
研究の問題過去30年間に何度か、プロトコルは「完了」しており、新しい研究トピックに目を向けるべきだと考えている人もいます。  
正式な手法は1990年代初頭に開花し、それから主要な管理プロトコルになった。 1990年代中頃、電子商取引メカニズムの提案が殺到し、忙しくなりました。  
アプリケーションプログラミングインターフェイス（API）のセキュリティを確認するために、もう1つはプロトコル分析ツールを適用しました  
多くのプロトコル研究は問題主導型ですが、まだ深い疑問があります。  
 そして、堅牢なプロトコルは一般にすべてが完全に指定およびチェックされるプロトコルであるという原則と、優れた仕様が実装者を過度に制約してはならないシステムエンジニアリングの原則との間の緊張をどのように管理しますか？  
歴史的背景については、元のニーダムシュローダー論文[1426]、バロウズアバディニーダム認証ロジック[357]、プロトコルの堅牢性に関する論文[2、112]、およびアンダーソンとニーダムによる調査論文[113]をお読みください。  
車のリモートキー入力やその他のセキュリティ問題については、ジープチェロキーをハッキングする方法についてのチャーリーミラーとクリスヴァラセックによる技術レポート[1316]が良い出発点です。