1はじめに暗号化は、セキュリティエンジニアリングと数学が出会う場所です。  
 これは分散システムを保護するための重要なテクノロジーですが、正しく行うことは驚くほど困難です。  
 残念ながら、利用可能な暗号化ツールは必ずしも非常に使いやすいとは限りません。  
 医学の友人はかつて、彼女は若い頃、経済的な理由から医学の学位を短縮し、専門家をできるだけ早く生産することに集中していた国で海外で働いていたと教えてくれました。  
外科医はファイルに尿検査がなかったという理由で劇場から患者を送り返しました。  
医師が生理学だけでなく手術を理解する必要があるのと同じように、セキュリティエンジニアは少なくとも暗号の基本（およびその他）に精通している必要があります。  
1つ目は、基礎となる直観で構成されます。これらの直観を明確にするために使用する2番目の数学、可能な限りセキュリティの証明を提供し、最も混乱を引き起こす構造を整理します。 3つ目は暗号工学です。これは、私たちが一般的に使用するツールであり、1495です。  
 歴史的背景彼らに何が悪いのかについての経験。  
 私はそれらを工学で説明し、必要なときに文献にアクセスできるように数学を十分にスケッチします。  
 たとえば、MicrosoftのCrypto API（CAPI）は、エンジニアが電子コードブックモードを使用するように微調整します。この章の終わりまでに、それが何であるか、なぜそれが悪いのか、そして代わりに何をすべきかを理解する必要があります。  
 暗号化とは、暗号を設計する科学と技術を指します。それらを解読する科学と芸術に対する暗号解読;暗号学は、しばしば単に暗号化に短縮されますが、両方の研究です。  
 その後、状況はやや複雑になります。  
 ブロック暗号は、共有鍵（秘密鍵または対称）と呼ばれる場合は暗号化と復号化の両方に1つの鍵を持っているか、公開鍵または非対称と呼ばれる場合は暗号化と復号化に別々の鍵を持っている場合があります。  
まず、基本的な概念を説明するために、いくつかの歴史的な例を挙げます。  
 最後に、より重要な暗号アルゴリズムが実際にどのように機能するか、およびそれらを使用してデータを保護する方法を示します。  
5。  
 アウグストゥスシーザーが王位を昇格したとき、彼は帝国の暗号システムを変更し、「C」が「A」、「D」が「B」などに書かれるようになりました。  
 驚くべきことに、同様のコードがシチリアのマフィアのカポディトゥッティカピであるベルナルドプロヴェンツァーノによって使用されました。彼は「4」を「a」、「5」を「b」などと書きました。  
アラブ人はこの考えをモノアルファベットの置換に一般化しました。ここでは、キーワードを使用して暗号アルファベットを置換します。  
1：abcdefghijklmnopqrstuvwxyzSECURITYABDFGHJKLMNOPQVWXZ図5。  
2。  
 トリックは、いくつかの文字、および文字の組み合わせが他のものよりもはるかに一般的であることです。英語で最も一般的な文字は、e、t、a、i、o、n、s、h、r、d、l、uの順です。  
 文字とダイグラム（文字のペア）の頻度のみを使用すると、通常、約600文字の暗号文が必要になります。予想される単語を推測するなどのよりスマートな戦略では、これを約150文字に削減できます。ニューラルネットワークを使用し、人間のアナリストの能力にアプローチする最先端のシステムも、UgariticやLinear Bなどの古代のスクリプトを解読することでテストされています[1194]。  
 前者では、暗号化ルールをプレーンテキストシンボルのストリーム内のプレーンテキストシンボルの位置に依存させる一方で、後でブロック内のいくつかのプレーンテキストシンボルを一度に暗号化します。  
2。  
 これは、 ‘A’ = 0、 ‘B’ = 1、…という規則を使用して、平文にキーを繰り返し追加することで機能します。  
。  
。  
、25]、つまり、[A 、.  
。  
 数学者はこれをasC = P + K mod 26と記述します。たとえば、P（15）をU（20）に追加すると35になり、26を引くと9になります。  
 この表記では、ジュリアスカエサルのシステムは固定キーK = Dを使用し、アウグストゥスカエサルのシステムはK = Cを使用し、Vigen`ereは繰り返しキー（ランニングキーとも呼ばれる）を使用しました。  
 テクノロジーが何であれ、キーに繰り返しキーワードを使用した暗号化は、図5のようになります。  
2 – Vigen`ere（ポリアルファベット置換暗号）女性化したGiacomo CasanovaからコンピューティングのパイオニアであるCharles Babbageまで、多くの人々がポリアルファベティック暗号の解決方法を考え出したようです。彼は、暗号文の十分に長い部分が与えられると、繰り返しパターンがキーワード長の倍数で現れることに気づきました。  
2。  
2、たとえば、「KIOV」は9文字後に繰り返され、「NU」は6文字後に繰り返されます。  
次に、暗号文の文字1、4、7などが同じキーレターですべて暗号化されました。したがって、頻度分析手法を使用して、この文字の最も可能性の高い値を推測し、残りのキーの文字に対してプロセスを繰り返すことができます。  
2。  
 これはワンタイムパッドとして知られており、第一次世界大戦中にギルバートバーナムによって提案されました[1001]。暗号文と同じ長さの平文があれば、暗号文を平文に復号化する鍵があります。  
 したがって、このシステムには完全な秘密があります。  
 あなたが「Heil Hitler」で始まっていることを知っていたwartimeGermanエージェントからのメッセージを傍受していて、暗号文の最初の10文字がDGTYI BWPJAであったと仮定します。  
3：PlainheilhitlerKeywclnbtdefjCipherDGTYIBWPJA図5。  
 図5に示すように、キーマテリアルがwggsb tdefjであるのと同じくらい簡単だったので、これも可能です。  
4 –スパイが主張できること今、私たちは暗号学では何も得られないことはめったになく、ワンタイムパッドの完全な秘密性の代償として、メッセージの完全性を完全に保護できません。  
4）：CipherDCYTIBWPJAKeywclnbtdefjPlainhanghitler図5。  
2。  
 実際、戦争中、クロードシャノンは、可能な平文として可能な限り多くのキーがあり、すべてのキーが等しく可能である場合にのみ、暗号が完全な秘密であることを証明しました。したがって、ワンタイムパッドは完全な秘密を提供する唯一の種類のシステムです。  
ワンタイムテープは、第二次世界大戦後期の双方によるトップレベルのコミュニケーション、その後NATO同盟国間の戦略的コミュニケーション、および1963年からのUS-USSRホットラインに使用されました。  
 ただし、このような暗号化は、トラフィックと同じ量のキーマテリアルを消費するため、ほとんどのアプリケーションにとって高額です。  
 次に、キーストリームを一度に1つのシンボルとデータと組み合わせることにより、データが暗号化されます。  
初期の例は、ローターマシン、非常に長い疑似ランダム状態のシーケンスを生成し、それらをプレーンテキストと組み合わせて暗号文を取得する機械的ストリーム暗号デバイスでした。  
 以下に説明する理由により、銀行は一般的に興味がありませんでしたが、WorldWar IIの戦闘員はローターマシンを使用して無線通信を暗号化し、同盟国がドイツの通信を解読するための手段には、コロッサスに関するアランチューリングと他の人々による研究は、戦後のコンピュータ業界のキックスタートを助けました。  
 ただし、ブロック暗号はより柔軟で、現在設計されているシステムではより一般的であるため、次にそれらを見てみましょう。  
2。  
ホイートストン暗号と呼ばれていないのは、政治家であるバロンプレイフェアにそれを示したからです。プレイフェアは夕食後にナプキンでアルバート王子とパーマーストン子爵（後の首相）にそれを示した。  
暗号博物館。  
米国で使用されているHagelinマシンの場合は2文字、ドイツのエニグマとイギリスのTypexSecurity Engineeringの場合は順列153ロスアンダーソン5。  
 歴史的背景キーワード、および「J」の文字の省略（図5を参照）  
6 – Playfair暗号化テーブル平文は、最初に「J」を「I」に置き換えて文字ペアに分割し、2つの文字が「x」で区切られるのを防ぎ、最後に「最後の文字のペアを完成させるために必要な場合はz '。  
次に、次のルールを使用して一度に2文字が暗号化されます。•2つの文字が同じ行または列にある場合、それらは後続の文字に置き換えられます。  
 たとえば、「lo」は「MT」に暗号化します。  
7 –プレイフェアの暗号化の例この暗号のバリアントは、第一次世界大戦ではイギリス軍によってフィールド暗号として使用され、第二次世界大戦ではアメリカ人とドイツ人によって使用されました。  
この場合も、ブロック暗号の出力が直感的に「ランダム」に見えるだけでは不十分です。  
 したがって、図5のキーを使用します。  
 一つの帰結は、十分な暗号文またはいくつかのありそうな単語が与えられると、テーブル（または同等のもの）が再構築できることです[740]。日本人がそれを傍受した場合、彼らはそれを解読した可能性があり、歴史は異なる可能性があります。  
2。  
1つの入力ビットを変更すると、平均して、出力ビットの半分が変更されます。  
ブロック暗号のセキュリティは、2文字より長いブロック長を選択することによっても大幅に改善できます。  
 以下では、DESとAESの内部の詳細について説明します。とりあえず、必要なのは単なる適切なブロックサイズだけではありません。  
 これにより、有効ではあるが無許可のトランザクションを生成するために、反対者が2つの異なる暗号文の一部をカットアンドペーストできるようになる可能性があります。  
 彼は兄弟に$ 1,000を配線し、銀行のコンピューターに「Pay John Smith $ 1,000」という別のトランザクションを挿入させ、この命令を傍受し、「Pay John Smith $ 10,000,000」として復号化した2つの暗号文から誤った命令を作成するかもしれません。  
5。  
4ハッシュ関数第3の古典的なタイプの暗号はハッシュ関数です。  
19世紀半ばに電信が発明された後、銀行は急速に主要なユーザーとなり、電子的に送金するためのシステムを開発しました。  
 younoで私たちのアカウントからお支払いください。  
 3012344567890123、これが「Doreen Smithからの結婚式のプレゼント」のためのものであることを彼に通知します。  
 弊社が負担する料金。  
19世紀、銀行、電信会社、海運会社は、トランザクションを保護するだけでなく、トランザクションを短縮できるコードブックを開発しました。これは、当時の国際電報のコストを考えると重要でした。  
 したがって、「当社のセキュリティエンジニアリング155ロスアンダーソン5からお支払いください。  
 あなたとの歴史的背景  
 時にはコードも暗号化されました。  
 たとえば、 '1000'のコードワードが 'mauve'で、 '1,000,000'のコードワードが 'magenta'の場合、コード化されたトラフィックを既知のトランザクションと比較できる曲がった電信担当者は、これを計算して、他の。  
 （現代の暗号学者はそれをハッシュ値またはメッセージ認証コードとして説明しますが、用語については後で詳しく定義します。  
 図5に示すように、銀行に支払い金額に対応する番号の表を備えたコードブックがあるとします。  
8 –シンプルなテストキーシステム今では、376,514ポンドのトランザクションを認証するために、重要な数字を無視して、53（100万なし）、54（300,000）、29（70,000）、71（6,000）を追加できます。  
ほとんどの実際のシステムはこれよりも複雑でした。彼らは通常、通貨コード、日付、さらには受取人の口座番号のテーブルを持っています。  
これにより、そのようなテストキーシステムは一方向の関数になりました。キーからの知識があれば、メッセージからテストを計算することは可能ですが、プロセスを逆にして、単一のテストからメッセージまたはキーを回復することはできません。テストには十分な情報が含まれていませんでした。  
 科学者のロバートフックは、1678年に分類されたアナグラム「ceiiinosssttuu」を公開し、2年後に「Ut tensio sic uis」から派生したことを明らかにしました。「力は緊張に応じて変化する」、または現在私たちは春のフックの法則と呼んでいます。  
）バンキングテストキーは、現代の暗号化の標準では強力ではありません。  
共犯者が銀行システムに慎重に選択したいくつかのメッセージを挿入すると、さらに簡単になります。  
2。  
 銀行のセキュリティコンサルタントとして数年働いて、昼食時に監査役の話を聞いていたとき、私はそれを悪用した2つの詐欺の事例を聞いただけです：攻撃者が銀行手続きを理解していなかったために失敗した暗号解読に関する外部の試みと、曲がりくねったsta↵メンバーが関与する成功した小規模詐欺。  
ただし、テストキーは、認証に使用される代数関数の歴史的な例です。  
 どちらの場合も考え方は同じです。一意のキーを使用して各メッセージを認証できる場合、単純な代数が理想的なセキュリティを提供します。  
 単に128ビットの素数pを選択し、A = k1M + k2（mod p）を計算できます。ここで、鍵は2つの128ビット数k1とk2で構成されます。  
 そのため、鍵についての知識がなければ、MとAを見る敵は、有効な偽造を作成するのに役立つ情報をまったく持っていません。  
 このユニバーサルハッシュ関数がブロック暗号でどのように使用されるか、および核のコマンドと制御でどのように使用されるかについては、パート2で説明します。2。  
 そのため、たとえば、現在ほとんどのWebサイトには、TLSと呼ばれるプロトコルを使用してセッションを暗号化できる公開キーを含む証明書があります。 Webページの所有者は、対応する秘密鍵を使用してトラフィックを復号化できます。  
これに関するいくつかのコンピューター前の例もあります。おそらく最高は郵便サービスです。  
 それが終わったら、私だけがそれを読むことができます。  
 同様のことが公開鍵暗号化でもうまくいかない可能性があります：偽の公開鍵がシステムに挿入されたり、コンピューターがハッキングされたり、人々が強制されたりする可能性があります。  
暗号化のもう1つの非対称アプリケーションは、デジタル署名です。  
3。  
 繰り返しになりますが、写本の署名や印鑑の形をしたコンピュータ以前の類似物があります。繰り返しますが、古いやり方と新しいやり方の両方で、失敗する可能性のある非常によく似たものがあります。  
3セキュリティモデル現代の暗号の詳細な設計に入る前に、さまざまなタイプの暗号と、それらのセキュリティを推論する方法を注意深く検討したいと思います。  
 私たちはすでに完全な秘密のモデルを見てきました。暗号文が与えられれば、その長さのすべての可能な平文は等しく可能性があります。  
2番目のモデルは具体的なセキュリティです。ここでは、敵対者が実行しなければならない実際の作業の量を知りたいです。  
 したがって、80ビットのキーでは、212倍、つまり約1か月かかります。デフォルトの最新システムである128ビットの鍵は、248倍困難です。  
 一般に、システムは、時間tで働いている敵が最大でifの確率で暗号を破ることに成功した場合、（t、✏）-安全です。  
 これにより、関心のある暗号の特定の性質について推論することができます。  
 これは、暗号学者と暗号解読者にゲームをプレイさせることによって形式化され、分析者は無視できない確率で識別できないはずの何かの効率的な弁別子を見つけて勝ちます。  
 通常、セキュリティの証明は、ランダム化（i。  
 確率的）nの時間多項式で実行されるアルゴリズムで、無視できない確率で情報を学習する場合、これにより、すでに信頼されている基礎となる暗号プリミティブの効率的な識別が可能になります。  
3。  
 これにより、KatzやLindell [1022]などの標準テキストで見つけられるかなりの数学的詳細がスキップされます。  
 暗号プリミティブをそのタイプのランダム関数と区別する効率的な方法がなく、特に、適用されるすべての統計的およびその他のランダム性テストに合格する場合は、暗号プリミティブを疑似ランダムと呼びます。  
図5。  
9）–私たちの写真では、サイコロと巻物で表されています。  
 もしそうなら、それはそこで見つけた答えを与えるでしょう。そうでない場合は、サイコロを投げてランダムに回答を生成し、将来の参照のために記録を保持します。  
 さらに、オラクルはいくつかの異なるルールに従って動作できます。  
3。  
 ランダム関数は、任意の長さの入力文字列を受け入れ、固定長の文字列、たとえばnbits長を出力します。  
 したがって、エルフには入力と出力の単純なリストがあり、それが機能するにつれて着実に成長します。  
3。  
 これらは、1960年代にパスワードの一方向暗号化のためにコンピューターシステムで最初に使用され、今日ではさらに多くの用途があります。  
 これらのハッシュ値は、ファイルが破損すると変更され、警察が証拠を改ざんしていないことを裁判所に保証できます。  
 ただし、ドキュメントがまだ秘密である場合（たとえば、優先日を設定したい発明など）、ドキュメント全体をアップロードするのではなく、メッセージハッシュのみをアップロードします。  
2。  
5。  
1。  
 入力xの知識があれば、ハッシュ値h（x）を簡単に計算できますが、そのような入力がまだ知られていない場合、h（x）でxを見つけることは非常に困難です。  
）出力はランダムなので、ランダム関数を反転させるために攻撃者が実行できる最善の方法は、幸運になるまで入力を増やし続けることです。 nビット出力の場合、これは平均で約2n1推測になります。  
 したがって、偶然に目的のターゲット出力を持つ入力を相手が推測できないほど多くの可能な出力がある場合、疑似ランダム関数も一方向関数になります。たとえば、SHA256が疑似ランダム関数であると主張する場合、特定の256ビット値にハッシュする入力を見つけるための実用的な方法はないと知っています。  
 したがって、秘密鍵kと計算h（x、k）を連結することにより、値xの一方向暗号化を取得できます。  
 （後で例22について説明します。  
1：1990年代から2000年代初頭にかけて多くの電話会社が携帯電話ユーザーの認証に使用したハッシュ関数はランダムではなく、攻撃につながりました。  
 対戦相手がショートカット攻撃（これは関数が疑似ランダムではなかったことを意味します）を見つけられない限り、衝突を見つける最善の方法は、大量のメッセージMiとそれに対応するハッシュh（Mi）を収集し、ハッシュをソートして探すことです試合。  
 この事実は、セキュリティエンジニアリングで非常に重要です160Ross Anderson5。  
 セキュリティモデルセキュリティエンジニアリングなので、詳しく見ていきましょう。  
3。  
2誕生日の定理誕生日の定理は、次の問題からその名前を取得します。  
ほとんどの生徒は直感的にそれはありそうにないと考え、次に数学の教師は生徒に次々に自分の誕生日を述べるように頼みます。  
 これはほとんどの人を驚かせるように、「誕生日のパラドックス」としても知られています。  
 湖にN匹の魚がいて、m匹を捕まえ、鳴らしてから投げ戻したとします。すでに鳴っている魚を最初に釣ったとき、mはNの約平方根になるはずです。  
この定理には、セキュリティエンジニア向けの多くのアプリケーションがあります。  
 これは、約2億の可能なペアが存在するためです。  
 ただし、主張された身元を確認するためにそれを使用することは問題ないかもしれません（他の多くのことが間違っている可能性があります。議論については、パート2の生体認証に関する章を参照してください）。  
たとえば、敵を識別するシステム（IFF）システムでは、一般的な機器の応答長は48〜80ビットです。  
しかし、衝突が許容できない他のアプリケーションがあります。  
 このようなアプリケーションでは、h（M1）= h（M2）で衝突を見つけることができたが、M1 = M2の場合、マフィアが所有する書店のWebサイトが適切なペアM1、M2を事前に計算し、M1に署名して「ラバーフェティッシュボリューム7のコピーを32ドルで注文します。  
、バミューダ。  
 。  
 （N�m+ 1）/ Nmこれは、N 'm2 / 2log（1 /�）[1037]によって漸近的に解決されます。  
3。  
 歴史的に、2つの最も一般的なハッシュ関数は、128ビットの出力を持ち、ブレークに最大264の計算を必要とするMD5と、160ビットの出力と最大280の暗号解読者の作業係数を持つSHA1でした。  
6。  
要約すると、暗号化ハッシュ関数が衝突耐性を持つ必要がある場合は、SHA-2やSHA-3など、少なくとも256ビットの出力を持つ関数を選択することをお勧めします。  
5。  
2ランダムジェネレーター–ストリーム暗号第2の基本的な暗号プリミティブはランダムジェネレーターで、キーストリームジェネレーターまたはストリーム暗号とも呼ばれます。  
 入力と出力が十分に長い優れた疑似ランダム関数がある場合、出力の数百ビットを除くすべてを破棄してハッシュ関数に変換し、数百ビットを除いてパディングしてストリーム暗号に変換できます。定数を使用して入力し、出力をキーストリームとして使用します。  
 （これは、排他的または加算モジュロ2であるため、加算的ストリーム暗号とも呼ばれます。  
 データを回復する必要がある場合は、ジェネレーターに戻り、同じキーを入力し、同じキーストリームを取得し、排他的または暗号化テキストを使用せずに、平文に戻します。  
 これはファイルの整合性を保証するものではないことに注意してください。ワンタイムパッドの説明で見たように、プレーンテキストにキーストリームを追加すると機密性を保護できますが、ファイルの変更を検出できません。  
 ファイル全体よりも変更からハッシュを保護する方が簡単な場合があります。  
 バーナムのオリジナルの電信暗号機は、パンチされた紙テープを使用していました。マークスは、SOEエージェントのシルクキーが、引退した婦人用カウンターでオックスフォードでどのように製造されたかを説明しています。物理的なセキュリティに関する章では、最新のハードウェア乱数ジェネレータについて説明します。  
3。  
 第二次世界大戦中、ロシアの外交交通量は、大使館に事前に配布した一時テープの量を超えたため、再利用されました。テキストメッセージには、実際には多くのデータを回復するのに十分な冗長性が含まれています。ロシアのトラフィックの場合、これにより、米国と英国が1943年以降の大量の戦時中のロシアのトラフィックを復号化し、数を分割したVenonaプロジェクトが発生しました。ロシアのスパイリング。  
これを回避するために、通常のエンジニアリング手法では、キーだけでなくシード（初期化ベクトルまたはIVとも呼ばれます）も使用するので、毎回異なる場所でキーストリームを開始します。  
 ここでは、古いキーストリームを再利用させようとする可能性のある敵対者がいる場合でも、両方の当事者が適切に機能するキーで同期することを確認する必要があります。  
3。  
 ここで、関数は反転可能であり、入力プレーンテキストと出力暗号テキストは固定サイズです。  
 シンボルの数と基になるアルファベットが何であれ、暗号化は固定長のブロックに作用します。  
）ブロック暗号化は以下のように可視化できます。  
 これは、左側に平文の列があり、右側に暗号文の列があります。  
 そうでない場合は、サイコロを転がして適切なサイズのランダムな暗号文を生成し（スクロールの右側の列にはまだ表示されません）、プレーンテキスト/暗号文のペアをスクロールに書き留めます。  
復号化を要求されると、エルフは同じことを行いますが、列の機能が逆になります。入力された暗号文を受け取り、それを右側のスクロールで探し、見つけたら、以前に関連付けられていたメッセージを表示します。  
ブロック暗号は、擬似ランダム置換のキー付きファミリーです。  
 各キーは、異なるスクロールに対応すると考えることができます。  
 さらに、誰もセキュリティエンジニアリング163ロスアンダーソン5。  
 セキュリティモデルは、まだ作成されていない平文または暗号文に関する情報を推測できる必要があります。  
 既知の平文攻撃では、標的キーに対応するオラクルからランダムに選択された多数の入力と出力が対戦相手に与えられます。  
 選ばれた暗号文攻撃では、彼はいくつかの暗号文クエリを作成します。  
最後に、関連するキー攻撃で、K + 1やK + 2などのターゲットキーKに関連するキーを使用して応答するクエリを作成できます。  
攻撃に関するこの精度は重要です。  
 多くの場合、そうではありませんが、メディアによって宣伝されています。そのため、上司や顧客に問題ではない理由を明確に説明できる必要があります。  
 たとえば、データ暗号化標準アルゴリズム（差分暗号解読）で発表された最初の主要な攻撃では、キーを回復するために247個の選択された平文が必要でしたが、次の主要な攻撃（線形暗号解読）では、既知の平文が243個に改善されました。  
 このような非実用的な攻撃は、実際的な悪用を提供するのではなく、暗号のセキュリティ認証に影響を与えるため、しばしば認証的と呼ばれます。  
 他のいくつかのケースでは、認証として開始された攻撃は、後のアイデアによって悪用に発展したものです。  
 たとえば、ブロードキャストエンターテインメントシステムを使用すると、ハッカーはデコーダーを購入し、多くの映画を見て、暗号化されたブロードキャスト信号と比較できます。そのため、既知の平文攻撃が主な脅威になる可能性があります。  
 歴史的な例は第二次世界大戦で、米国のアナリストは、彼らが疑わしい島「AF」に対する日本の意図をミッドウェイを意味すると知った。  
 ミッドウェイが日本の目的であることを知って、チェスターニミッツ提督は彼らを待っていて、4隻の日本の空母を沈め、戦争の流れを変えました[1001]。  
3。  
 選択された平文/暗号文攻撃は、脅威が昼休みの攻撃である場合に心配になる可能性があります。許可されたユーザーが不在の間に暗号デバイスに一時的にアクセスし、許可された操作の全範囲を選択したデータでしばらく試行します。  
このような攻撃をすべて排除するための目標は、前述のようにセマンティックセキュリティです。暗号化では、無視できる確率以外で、無許可の情報（プレーンテキスト、暗号化テキスト、またはキー）を推測できません。  
3。  
 私たちの類推を続けると、ユーザーは彼女とエルフだけが知っているスクロールに秘密の名前を付け、エルフの公開一方向関数を使用してこの秘密の名前のハッシュを計算し、ハッシュを公開して、暗号化操作を実行するようエルフに指示することができますこのハッシュを引用する人のために。必要なのは、彼らがオラクルにアクセスできることだけです。  
 これは誰でも実行できる計算ですが、秘密鍵などのトラップドアを知っている人だけが元に戻すことができます。  
 それにもかかわらず、それを正式に述べましょう：公開鍵暗号化プリミティブは、ランダム入力Rが与えられると、プロパティthat1を持つ2つの鍵、KR（公開暗号化鍵）とKR�1（秘密復号鍵）を返す関数で構成されます。  
 暗号化機能があります{。  
 。  
 復号鍵KR�1を使用して暗号文Cに適用される復号化関数があり、元のメッセージM = {C}KR�1を生成します。  
ほとんどの実際のシステムでは、暗号化はランダム化されているため、誰かが同じ公開鍵を使用して同じメッセージを暗号化するたびに、答えは異なります。これはセマンティックセキュリティのために必要です。そのため、対戦相手は、特定の暗号文の平文の推測が正しいかどうかを確認できません。  
3。  
 しかし、今のところこれで十分です。  
3。  
 基本的な考え方は、メッセージの署名は1人のプリンシパルだけが作成でき、だれでもチェックできるというものです。  
 アプリケーションにはソフトウェアの更新の調整が含まれるため、PCはWindowsの更新が外国の諜報機関ではなくMicrosoftによって実際に作成されたものであると判断できます。  
 （後者はより手書きの署名に似ています。2つが同じになることはありませんが、銀行には、所定の標本が本物か偽造かを判断する手段があります。  
 その場合、署名が与えられれば、誰でもそれが生成されたメッセージを回復できます。そうでない場合、検証者は検証を実行する前にメッセージを知っているか推測する必要があります。  
 公開署名検証鍵V Rが与えられると、秘密署名鍵�R; 2を計算することは不可能です。  
 署名Sig�R{M}と公開署名検証キーV Rが与えられた場合、署名が�Rで正しく計算された場合はTRUEを出力し、それ以外の場合はFALSEを出力する検証関数があります。  
 他の方向では、誰に対しても検証を実行します。  
 しかし、メッセージの回復を伴うスキームでは、誰でも署名を入力して、それに対応するメッセージを取り戻すことができます。  
4。  
 これは望ましい場合があります。低帯域幅チャネルを介して短いメッセージを送信する場合、署名とメッセージではなく署名のみを送信すれば、スペースを節約できます。  
 これについては、セクション16の最後で説明します。  
2。  
 ハッシュ関数は一方向だけでなく、衝突耐性も必要です。  
4対称暗号アルゴリズム定義を整頓したので、内部を調べて、実際にどのように実装できるかを確認します。  
 実際、研究レベルでも、ほとんどの暗号化は数学と同じくらいコンピュータサイエンスです。現代の暗号攻撃は、ビットの推測、パターンの検索、可能な結果の並べ替えなどから組み立てられ、特に高額なものではなく、工夫と永続性が必要です。  
4。  
 たとえば、いくつかのキーマテリアルを入力テキストのブロックに追加してから、入力のサブセットを表示し、この方法で何度も続けることができます。  
 ブロック暗号には混乱と混乱が必要です。  
 図5。  
この配置のポイントは、デジタルロジックに任意の16ビットから16ビットの機能を実装する場合、220ビットのメモリが必要であり、各単一の出力ビットに対して216ビットのルックアップテーブルが1つ必要です。  
4。  
10：–単純な16ビットSPネットワークブロック暗号化ゲート数。4ビットから4ビットの機能は、4 x 24または64ビットのメモリしか必要としません。  
 キーは、いくつかの4ビットSボックスのいくつかの選択で構成される場合があります。または、混乱を提供するために各ラウンドに追加され、Sボックスを介して供給されるテキストが拡散を提供します。  
 暗号は十分に「広い」必要があります2。  
 Sボックスは適切に選択する必要があります。  
4。  
1ブロックサイズ最初に、16ビットブロックで動作するブロック暗号は、対戦相手が平文と暗号テキストブロックの辞書を作成するだけで構築できるため、かなり制限されます。  
 したがって、実用的なブロック暗号は通常、平文と64ビット、128ビット、またはそれ以上の暗号文を扱います。  
5。  
1。  
 図5の2つのラウンド。たとえば、右端の12ビットを一定に保ち、左端の4ビットを微調整して、左上のSボックスの値を推定することができます。  
4。  
しかし、それはまだ基本的な学生の練習です。  
 単純な例では、Sボックスの1つのラウンドからの各出力ビットが次のラウンドの1つの入力ビットにのみ接続されるため、拡散は非常に遅くなります。  
 暗号化だけでなく復号化にもブロック暗号を使用する場合、この線形変換は可逆でなければなりません。  
5。  
1。  
S-boxが入力（0,1,2 ,.  
。  
 次に、入力の最も重要なビットは、出力の最も重要なビットとして変更されずに渡されます。  
 私たちの暗号が疑似ランダムであるとは断言できませんでした。  
4。  
4Linear Cryptanalysis実際のブロック暗号の攻撃は、通常、この例よりも見つけるのが困難ですが、同じ考え方を使用しています。  
 線形暗号解析[895、1244]は、「最初のSボックスへの入力のビット2とビット5が出力のビット1とビット8に等しく、確率13/16で等しい」などの関係を収集し、次にそれらを一緒に接着して、入力ビット、出力ビット、および半分とは異なる確率で保持されるキービットの間の代数的関係に転送します。  
5 + 1 / M、確率理論のサンプリング定理によれば、約M 2の既知のテキストが得られると、キービットの回復を開始することが期待できます。  
5。  
1。  
 8ビットSボックスの典型的な観察は、「weSecurity Engineering169Ross Anderson5の場合」です。  
 SYMMETRIC CRYPTO ALGORITHMS ﬂ ip入力ビット2、3、および7を一度に入力すると、確率11/16で、ipを出力する唯一の出力ビットは0および1になります。  
 分析手順は、考えられるすべての入力差異パターンを調べ、それらの値�i、�oを探して、�iの入力変化が特に高い（または低い）確率で�oの出力変化を生成するようにします。  
十分に選択された入力があれば、予想される出力が表示され、キーについて推論することができます。  
 （32ビットブロックの暗号と、単一ペアが2-40の場合の成功確率が差分攻撃の128ビットキーである場合など、病理学的なケースには注意する必要があります。  
）これら2つのテーマには多くのバリエーションがあります。  
 これは不可能暗号解読という魅力的な名前を持っていますが、多くのシステムに対して非常に完全に可能です[242] 4。  
 たとえば、ラウンドを注意深く設計することにより、ラウンドごとの情報漏えいを減らし、必要なラウンド数を減らすことができます。  
 単純なラウンドも分析が容易な場合があります。  
 しかし、暗号が私たちが知っているすべての攻撃に抵抗し、ある程度の安全マージンがあることを示すことはできるかもしれませんが、これはそれが新しいタイプの攻撃に抵抗するかどうかについてはほとんど言いません。  
）5。  
2高度暗号化標準（AES）高度暗号化標準（AES）は、発明者のVincent RijmenおよびJoan DaemenにちなんでRijndaelとして最初に知られているアルゴリズムです[507]。  
 これはSPネットワークです。それを指定するには、Sボックス、ラウンド間の線形変換、およびキーを計算に追加する方法を固定する必要があります。  
実装目的では、これは単に4のルックアップテーブルと見なすことができます。これは、第二次世界大戦のブレッチリーで最初に使用された可能性があります。  
4。  
 この構成は、厳密な微分境界と線形境界を与えます。  
 最初のステップはshu✏eです。4バイトの一番上の行は変更されずに残り、2番目の行は左に1桁、3番目の行は2桁、4番目の行は3桁シフトされます。  
 これを図5に示します。  
この組み合わせの影響は、暗号への入力の変更が2ラウンド後にすべての出力に影響を与える可能性があることです。雪崩の影響により、線形攻撃と微分攻撃の両方が困難になります。  
 。  
1234Shift�row1423Mix�column図5。  
 これは、ラウンドごとに16バイトの鍵素材が必要であることを意味します。それらは、反復関係を使用して、ユーザーが提供したキーマテリアルから派生します。  
 実際には、AESの競争時に予想していたとおり、およびこの章の以前の版で説明したように、これらは実用的なセキュリティを提供するのに十分です。ブルービットフォース検索の2127および2255とは対照的に、256ビットAESの場合は3。  
 しかし、これらの攻撃は、実行不可能なほど大量のテキストまたは関連するキーの非常に特殊な組み合わせを必要とするため、実際には違いはありません。  
 （Camelliaは、自身のAES候補者が最初のラウンドでノックアウトされたチームによって設計されました。  
4。  
 さて、私はAESファイナリストSerpent [94]の設計者の1人で、コンテストで2位になりました。勝者であるRijndaelが86票、Serpent 59票、Twofish 31票、RC6 23票、MARS 13票を獲得しました。  
10、しかし十分に広く、十分なラウンドを持つように修正され、現在出現している攻撃を見越して、Rijndaelよりはるかに大きなセキュリティマージンを持つように設計されました。  
プロセス全体に関与し、1990年代の大半で共有キー暗号の分析と設計に取り組んできたので、AESは、数学的暗号分析に基づく実際の攻撃に対して安全であるという高いレベルの信頼を得ています。  
 実際の攻撃には、タイミング分析と電力分析が含まれます。  
 後者の場合、対戦相手は、暗号を実行するデバイスによって引き出される電流の測定値を使用します。顧客がマフィア所有の店の端末に置く銀行のスマートカードを考えてください。  
 「AESが壊れた場合に備えて」Serpentを実装することも意味がありません。スワップ可能なアルゴリズムを持つことは、プラグイン可能な暗号法として知られていますが、アルゴリズムネゴシエーションプロトコルでの致命的なエラーのリスクは、だれでも来るリスクよりも桁違いに大きいですAESへの本番攻撃に対処します。  
）裏話は、1970年代に、NSAが以前の標準ブロック暗号であるData EncryptionStandard（DES）の選択とパラメータを操作して、当時の米国の業界にとって十分な暗号を提供するようにしたことです。外国政府にそれが安全でないと信じさせている間、彼らは代わりに彼ら自身の弱いデザインを使いました。  
 AESはこの脚本をフォローしているようです。数学的にだけ十分に強力で、その安全な実装にスキルと注意が必要なアルゴリズムを選択することにより、米国政府は、ロシア、中国、日本などの企業が、スキルとエフォートが低いため安全性の低いシステムを使用することになると考えました実装に投資しています。  
 ああ、そしてNSAは、2005年以降、SECRETまでの情報を保護するための128ビットキーと、TOPSECRET用の192ビットまたは256ビットキーを備えたAESを承認しています。  
AESの明確な仕様は連邦情報処理標準197であり、その発明者はその設計を詳細に説明した本を書いている[507]。  
4。  
4。  
 その後、FeistelはIBMに移り、データ暗号化標準（DES）アルゴリズムを作成する研究グループを設立しました。これは、依然として支払いシステムのセキュリティの柱です。  
12。  
 一部のFeistel暗号ではキャリー付き加算も使用されますが、左半分の丸め関数f1が計算され、排他的論理和（キャリーなしのバイナリ加算）を使用して右半分と結合されます。  
）次に、右半分の関数f2が計算され、左半分と結合されます。  
Feistel暗号で表示される表記は（f、g、h、。  
。  
。  
 連続したラウンド関数です。  
。  
f2k�1、f2k）。  
。  
、f2k�1、f2k）=（f2k、f2k�1、。  
。  
 したがって、ラウンド関数fiは可逆である必要はありません。Feistelstructureを使用すると、一方向関数をブロック暗号に変換できます。  
5。  
3。  
 彼らは、fiがランダム関数である場合、（f1、f2、f3）は、選択された平文攻撃の下でのランダム置換と区別がつかないことを示し、この結果はすぐに拡張され、（f1、f2、f3、f4）が選択された平文/暗号文攻撃–言い換えると、これは疑似ランダム置換でした。  
）工学用語では、効果は非常に良いラウンド関数が与えられた場合、4ラウンドのFeistelで十分です。  
5。  
3。  
広く導入された「キラーアプリ」はATMネットワークでした。そこからセキュリティエンジニアリング173ロスアンダーソン5。  
 SYMMETRIC CRYPTO ALGORITHMS ??XXXXXXXXXXXXXXXXXXXX⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠⇠�•f2k��?.  
 。  
12：– Feistel暗号構造Security Engineering174Ross Anderson5。SYMMETRIC CRYPTO ALGORITHMSこれは、前払いメーター、交通機関のチケットなどに広がります。  
 そのラウンド関数は32ビットハーフブロックで動作し、3つの演算で構成されます。最初に、ブロックが32ビットから48に拡張されます。次に、48ビットのラウンドキーが排他的論理和を使用して混合されます。•結果がパススルーされます。 8つのSボックスの行。それぞれが6ビットの入力を受け取り、4ビットの出力を提供します。最後に、出力のビットは、固定パターンに従って並べ替えられます。  
13：Si – 1Si + 1Keyがここに追加されました•••�•••�Si図5。  
 DESの完全な仕様は[1397]に記載されています。  
 最も説得力のある批判は、キーが短すぎるということでした。  
 WhitDi�eとMartin Hellmanは1977年に、DESキーサーチマシンは100万のチップで構築でき、それぞれが毎秒100万のキーをテストすると主張しました。 100万は約220なので、キーを見つけるのに平均で215秒、つまり9時間以上かかります。  
 科学者がDESを発明したIBMは、そのようなマシンを構築するために米国政府に2億ドルを請求することになると反論しました。  
）1980年代の間、DESキーサーチマシンがさまざまな諜報機関によって構築されたという噂が根強く残っていましたが、最初の成功した公開キー検索攻撃は1997年に行われました。  
 1998年、Electronic Frontier Foundation（EFF）は、Deep Crackと呼ばれるDESキーサーチマシンを250,000ドル未満で構築し、3日でDESチャレンジを破りました。  
4。  
 したがって、検索率は2でした。  
 クラッカーのデザインは公開されており、[619]にあります。  
 100,000台のマシンを備えた最新のボットネットは数時間かかります。  
DESに対するもう1つの批判は、IBMが設計原則を米国政府の要請に応じて秘密にしていたため、簡単にアクセスできる「罠」があったということです。  
 この話は、IBMが1972年にこれらの技術を発見したこと、そして米国の国家安全保障局（NSA）がさらに以前に発見したことでした。  
 これらすべての政治的側面については、26で説明します。  
7。  
これで、DESをかなり完全に分析できました。  
 DESは20ラウンド以上で安全ですが、実用上、そのセキュリティはキーの長さによって制限されます。  
 したがって、既知のショートカット攻撃は問題ではありません。  
 AESと同様に、タイミング分析と電力分析に基づく攻撃もあります。  
 銀行ネットワークは、1999年以降の標準であるTriple-DESに大きく移行しています[1397]。  
 形式的に：3DES（k0、k1、k2; M）= DES（k2;DES�1（k1; DES（k0; M）））3つのキーを等しく設定すると、単一のDESencryptionと同じ結果が得られるため、レガシー機器との下位互換モード。  
）ほとんどの新しいシステムはデフォルトの選択としてAESを使用しますが、ATM、POS端末、および銀行ネットワークがそれぞれに通信する多くのプロトコルで使用されるメッセージ形式のため、多くの銀行システムは8バイトブロックのブロック暗号の使用に専念しています。その他、および顧客のPINを生成して保護するためにブロック暗号を使用しているため（これについては、バンキングと簿記の章で説明します）。  
キーサーチを防ぐ別の方法（および電力分析を難しくする方法）は、ホワイトニングです。  
 この複合暗号は、DESXとして知られています。  
5。  
 ホワイト化されたブロック暗号は一部のアプリケーションで使用されます。具体的には、以下で説明するXTS操作モードで使用されます。  
5。  
5。  
 これは、固定ブロックサイズ（DESの場合は8バイト、AESの場合は16バイト）のブロック暗号を拡張して、任意の長さのメッセージを処理する方法を指定します。  
 それらを理解することは非常に重要であり、特にいくつかの一般的なツールはデフォルトで弱いツールを提供するため、ジョブに適したものを選択できます。  
5。  
1ブロック暗号を使用しない方法電子コードブックモードでは、上のPlayfairの例のように、プレーンテキストの後続の各ブロックをブロック暗号で暗号化して暗号文を取得します。  
しかし、冗長データを暗号化するためにそれを使用すると、パターンが透けて表示され、プレーンテキストに関する対戦相手の情報が提供されます。  
14は、ECBmodeでDESを使用して暗号化された場合に漫画の画像がどうなるかを示しています。  
前世紀の1つの人気のある企業メールシステムで使用された暗号化は、8文字のパスワードから派生したキーを使用したDES ECBでした。  
 これは、フィールドDES暗号化システムに対してこれまでで最も単純な攻撃の1つを与えました：辞書の各パスワードでヌルブロックを暗号化し、回答をソートします。さらに、ECBモードを使用して、銀行の支払いメッセージなど、信頼性を必要とする複数のブロック長のメッセージを暗号化することは、ブロック境界に沿ってカットアンドスプライス攻撃を開始するため、特にばかげています。  
セキュリティエンジニアリング177ロスアンダーソン5。  
 動作モード（a）平文（b）ECB暗号文図5。  
5。  
2暗号ブロックチェーン暗号ブロックチェーン（CBC）モードを使用するために使用される複数のブロックを暗号化するほとんどの商用アプリケーション。  
 その中で、暗号化する前に、暗号文の前のブロックを平文の現在のブロックに排他的に（または図5を参照）します。  
このモードは、平文のパターンを偽装します。各ブロックの暗号化は、前のすべてのブロックに依存します。  
ただし、平文の一部を知っている相手は、メッセージ（または同じ鍵で暗号化されたいくつかのメッセージの一部）を切り取って接合することができる場合があります。  
 そのため、CBC暗号化は通常、別の認証コードで使用する必要があります。  
 これは2002年にSerge Vaudenayによって発見されました。その変種は、遅くとも2016年[1949]にSSL、IPSECおよびTLSに対して使用されていました。  
5。  
。  
図5。  
5。  
 それらを並列化するのは困難です。  
 これは、バックボーンリンク上のトラフィックの保護など、高速アプリケーションでは不便な場合があります。  
最も簡単な解決策は、AESをストリーム暗号として使用することです。  
セクション5で説明したように、加法ストリーム暗号には2つの体系的な脆弱性があります。  
上記2。  
 2つ目は、メッセージの整合性を保護できないことです。  
 これらのメッセージは高度に構造化されています。たとえば、37〜42バイト目に転送される合計が含まれていることを知っているかもしれません。  
あなたは銀行に入り、500ドルをanaccompliceに送ります。  
 バイト37–42のMiを知っているので、Kiを回復して、受信銀行に500ドルではなく500,000ドルを支払うように指示する変更されたメッセージを作成できます。これは、詳細な攻撃の例です。これは、1回限りのパッドから得られる完全な秘密の価格だけでなく、より単純なストリーム暗号の価格でもあります。  
5。  
5。  
4レガシーストリーム暗号モード2つの古いストリーム暗号操作モード、出力フィードバックモード（OFB）と頻度の低い暗号文フィードバックモード（CFB）を見つけることができます。  
 初期化ベクトルのIVを書くと、K1 = {IV} KおよびKi = {IV} K（i�1）になります。  
 したがって、高速リンクでトリプルDESなどの64ビットブロック暗号を使用する場合、サイクル長の問題が発生する可能性があります。232を少し超える疑似ランダム64ビット値を呼び出すと、確率が一致するようになります。  
）ただし、カウンターモードの暗号化は、2n / 2ではなく2nのサイクル長が保証されており、前述のように並列化は簡単です。  
暗号フィードバックモードは、妨害に抵抗する必要がある無線システムで使用するために設計された、別の種類のストリーム暗号です。  
 これは、ブロック暗号を使用して暗号文の最後のnビットを暗号化し、最後の出力ビットを次の平文ビットに追加し、暗号文を1ビットに沿ってシフトすることによって実現されます。  
5。  
5メッセージ認証コードブロック暗号の別の正式な動作モードは、データの暗号化には使用されませんが、データの整合性と信頼性を保護します。  
 ブロック暗号を使用してメッセージのMACを計算するには、CBCモードを使用して暗号化し、最後のものを除くすべての出力暗号文ブロックを破棄します。この最後のブロックはMACです。  
）この構造により、MACはすべての平文ブロックと鍵に依存します。  
メッセージの長さが可変である場合、1つの文字列で計算されたMACが、異なる文字列でMACを計算するためのIVとして使用できないことを確認する必要があります。これにより、相手がMACを取得して不正を行うことができなくなります。 2つの構成SecurityEngineering180Ross Anderson 5。  
 動作モード文字列。  
 （CMACは、岩田哲と黒澤薫[965]の提案に基づいています。  
MACには他にも考えられる構造があります。最も一般的なものは、キー付きのハッシュ関数を使用するisHMACです。セクション5で説明します。  
2。  
5。  
 整合性と機密性を保護する必要がある一括暗号化には効率的ではありません。 CBCモードまたはカウンターモードのいずれかを使用してデータを暗号化し、CBC-MACまたはCMACを使用してその整合性を保護する場合、処理するデータの各ブロックに対してブロック暗号を2回呼び出し、操作を並列化できません。Galois Counter Mode（GCM）は、2007年にNISTによって承認されて以来、デフォルトとして引き継がれました[1407]。  
暗号化は、カウンターモードのバリアントで実行されます。結果の暗号文は、多項式の係数としても使用され、2128要素のガロアフィールド上のキー依存ポイントで評価されて、認証子タグを提供します。  
2。  
 提供されたキーは、ランダムなIVとともに使用され、一意のメッセージキーと一意の認証キーの両方を生成します。  
GCMには興味深い増分プロパティもあります。新しいオーセンティケーターと暗号文は、変更されたビット数に比例した量で計算できます。  
 GCMは、バルクコンテンツの認証済み暗号化のデフォルトの設定です。  
0以降;これはカウンターモードとCBC-MACを組み合わせるので、計算にかかる時間の約2倍のコストがかかり、並列化したり、段階的に再計算することはできません[1406]。  
5。  
 これは、プレーンテキストの長さを維持する操作モードを好むハードディスク暗号化などのアプリケーションでは非常に不便です。  
6。  
16：–フィードフォワードモード（ハッシュ関数）の動作モード、XTS-AES、GCMに触発され、2007年に標準化されました。  
 形式的には、ブロックMiでキーKを使用して暗号化されたメッセージMi isAESX（KTj、K、KTj; M）で、微調整キーKTjは、異なるキーを使用してIVを暗号化し、適切な定数で繰り返し乗算することによって導出されます。各ブロックに異なるホワイトナーを与えるため。  
 暗号文の操作を検出するために上位層のメカニズムが依然として必要ですが、単純なチェックサムで十分です。  
セクション5のハッシュ関数  
3。  
ブロック暗号からハッシュ関数を構築することも可能です5。  
 この操作を不可逆にするために、フィードフォワードを追加します。（i�1）番目のハッシュ値は、ラウンドiの出力と排他的または加算されます。  
16）。  
セキュリティエンジニアリング182ロスアンダーソン5。  
 ハッシュ関数誕生日定理はここで別の形をしています。ハッシュ関数hがnビットブロック暗号を使用して構築されている場合、2つのメッセージを見つけることが可能です。M1= M2 with h（M1）= h（M2）with about 2n / 2e↵ ort（その多くのメッセージMiよりも少し多くハッシュして、一致を探します）。  
 AESなどの128ビット暗号は、ほぼ適切なものでした。実際、Blu-ray DVDのAACSコンテンツ保護メカニズムは、このようにAESから派生したハッシュ関数である「AES-H」を使用していました。  
6。  
 最初の2つはRon Rivestによって設計され、他の2つはNSAによって設計されました。•MD4には3つのラウンドと128ビットのハッシュ値があり、1998年に衝突が見つかりました[568]。•MD5には4つのラウンドと128ビットのハッシュ値、 2004年に衝突が発見された[1979年、1981年];•1995年にリリースされたSHA-1には、5つのラウンドと160ビットのハッシュ値があります。  
これらのハッシュ関数の基礎となるブロック暗号は似ています。それらのラウンド関数は、32ビットプロセッサで使用可能なレジスタ操作の複雑な混合です[1667]。  
 MD4は1998年にハンス・ドベルティンによって破られました[568]。 MD5は、2004年にXiaoyun Wangとその同僚によって破られました[1979年、1981年]。意味のあるテキストを含み、デジタル証明書に使用されているようなメッセージ形式に準拠している文字列の間でも、衝突を簡単に見つけることができます。  
 2017年2月、アムステルダムとグーグルの科学者が1つを発表し、要点を証明し、人々にSHA-2 [1828]（および以前のバージョンのTLSからTLS 1）などのより強力なハッシュ関数に移行するように説得しました。  
2020年に、ガセタンルーレントとトーマスペイリンは、選択されたプレフィックスの衝突を計算する改善された攻撃を開発し、数万ドルのコストで認証を可能にしました[1146]。  
 受賞者のKeccakは内部構造がかなり異なり、2015年にSHA-3として標準化されました。  
セキュリティエンジニアリング183ロスアンダーソン5。  
 ハッシュ関数導入されたシステムの多くは、MD5などのハッシュ関数を使用しています。これには、簡単な衝突検索アルゴリズムがあります。  
 警察が証拠を改ざんしていないことを裁判所に再確認するために、押収されたコンピューターにファイルのハッシュを保持するフォレンジックシステムについてはすでに触れました。ハッシュの衝突は、警察か被告かを問わず、誰かが改ざんしようとしていることを示すだけで、より慎重な調査をトリガーします。しかし、銀行システムはそのようには機能しません。  
 どちらの場合も、おそらく既存のハッシュ値のプリイメージを見つける必要があります。これは、衝突を見つけるよりもはるかに難しい暗号解読タスクです。  
6。  
 法医学的な作業を行う一部の人々は、MD5を何年も使用しており、その衝突は有用な攻撃を与えないため、MD5を使用し続けます。  
2005年、オーストラリアのシドニーでスピード違反を犯したと非難された運転手は、ニューサウスウェールズ州の道路と交通局がMD5がこのアプリケーションで安全であると証言する専門家を見つけられなかったため、無罪となりました。  
 そのため、脆弱性がエンジニアリングの脅威を提示していなくても、認​​証の脅威を提示する可能性があります。  
 それらの1つはMACを計算することです。  
ただし、HMACと呼ばれるこれを行うための受け入れられた方法は、この計算の結果が再度ハッシュされる追加のステップを使用します。  
 したがって、HMACk（M）= h（k�B、h（k�A、M））です。  
 ハッシュ関数が弱い側にある場合、この構造は悪用可能な衝突を見つけにくくする可能性があります[1089]。  
ハッシュ関数のもう1つの使用法は、後で明らかにすることを約束することです。  
 その場合、ドキュメントのハッシュを公開したり、商用のタイムスタンプサービスに送信したり、ビットコインブロックチェーンにマイニングしたりできます。  
繰り返しますが、タイムスタンプを行うときにペアを渡す必要があるため、衝突ペアを生成するアルゴリズムはこれを壊しません。  
7。  
 入力は、ツリーの葉を形成する値にハッシュされます。各非リーフノードには子ノードのすべてのハッシュのハッシュが含まれているため、ルートのハッシュはリーフのすべての値のハッシュです。  
また、ブロックチェーンアプリケーションでも広く使用されています。実際、ブロックチェーンは単なるマークルツリーです。  
 たとえば、Lamportデジタル署名はハッシュ関数から構築できます。ランダムな512ビットの値kiの秘密鍵を作成し、検証キーVをマークルツリーハッシュとして公開します。  
 これは、ハッシュ関数が安全であれば安全ですが、各キーは1回しか使用できないという欠点があります。  
 ただし、ほとんどの目的で、人々は数論に基づく署名アルゴリズムを使用します。これについては、次のセクションで説明します。  
 キーの更新とは、キーを共有する2つ以上のプリンシパルが、合意された時間で一方向ハッシュ関数にキーを渡すことを意味します：Ki = h（Ki�1）。  
 妥協の連鎖は、ハッシュ関数の一方向性によって破られます。  
 バリアントは自動キーイングであり、プリンシパルは最後のキー変更以降に交換したメッセージでキーをハッシュしてキーを更新します。Ki+ 1 = h（Ki、Mi1、Mi2、。  
 。  
 攻撃者がシステムの1つを侵害してキーを盗んだ場合、攻撃者が観察または推測できないメッセージを交換するとすぐに、セキュリティが回復します。繰り返しになりますが、妥協の連鎖は壊れています。  
 オーストラリアのEFT決済端末のバンキングで最初に使用されました[207、209]。  
 これがどのように機能するかを次に説明します。  
7非対称暗号プリミティブ非対称暗号、公開鍵暗号、およびデジタル署名で一般的に使用される構成要素は、数論に基づいています。  
基本的な考え方は、暗号のセキュリティを、多くの人が解こうとして失敗したという意味で、難しいことがわかっている数学的な問題を解く難しさに依存させることです。  
7。  
5。  
1素因数分解に基づく暗号化素数は、適切な除数のない正の整数です。素数を分割する数は、1と数そのものです。  
。  
}。  
 素数を見つけてそれらを掛け合わせて合成数を得るのは簡単ですが、合成数をその因子に分解するのははるかに困難です。  
 2020年に因数分解される2つのより大きな素数素数の最大の複合積は、829ビットの数値（10進数で250桁）のRSA-250でした。  
2GHzコア;以前の記録である2019年のRSA-240は、900年に相当しました[301]。  
 1024ビットの数値については、NSAがすでにそれらを因数分解できると思います。第2版では、「因数分解の履歴の外挿から、最初の因数分解が2018年に公開されることを示唆しています。  
 とにかく、キーを何年もの間安全に保ちたい組織は、少なくとも2048ビットの数値をすでに使用しています。  
 これは、フェルマーの小さな定理を使用しています。これは、すべての素数pがaを分割しないことを示しています。ap�1⌘1（mod p）（証明：セット{1、2、。。  
 一般的な整数nの場合、a�（n）⌘1（mod p）であり、オイラーの関数�（n）はnより小さい正の整数の数であり、除数が共通ではありません（証明は類似しています）。  
RSAでは、暗号化キーは因数分解が困難な係数N（ランダムに選択された2つの大きな素数pとq、たとえばそれぞれ1024ビットのtakeN = pq）と、p�1のいずれにも共通の因子がない公開指数eです。またはq�1。  
 MがメッセージでCが暗号文である場合、暗号化はCによって定義されます。⌘M e（mod N）復号化は逆の操作です。M⌘epC（mod N）秘密鍵を知っている誰でも– Nの因子pおよびq –簡単にepCを計算できます（mod N）。  
7。  
epC（mod N）はCd（mod N）として計算されるようになり、解読はフェルマーの定理のために機能します。Cd⌘{M e} d⌘M ed⌘M 1 +k�（N）⌘M。  
1⌘M（mod N）同様に、秘密鍵の所有者はそれを使用してメッセージを操作し、signatureSigd（M）⌘M d（mod N）を生成でき、この署名は、指数e mod Nに上げることで検証できます。 （したがって、eとNを公開署名検証キーとして使用します）、メッセージMが復元されたことを確認します。M⌘（Sigd（M））e（mod N）RSA暗号化も署名も、そのままでは安全ではありません。  
 たとえば、M1M2 = M3のようなプレーンテキスト間で保持される関係がある場合、同じ関係が暗号文C1C2 = C3と署名Sig1Sig2 = Sig3間で保持されます。  
 生のRSAの準同型の性質は、公開鍵の暗号化または署名のランダムなオラクルモデル定義に適合しないことを意味します。  
RSAでは、代数的攻撃につながる可能性があるため、小さな指数eを使用して同じメッセージを複数の受信者に暗号化することも危険です。  
 同じ短いメッセージを「攻撃」と暗号化するたびに、完全に異なる暗号文を取得し、これらを互いに区別したり、「後退」のための暗号文から区別したりしたいと考えています。  
暗号理論家は、非対称暗号で問題が発生する可能性のあるすべての問題を分析し、それを整理する方法を見つけるために何十年も取り組んできました。  
 つまり、暗号化には、選択された平文攻撃だけでなく、選択された暗号文攻撃にも対抗する必要があります。  
7。  
通常の実際のソリューションは、最適な非対称暗号化パディング（OAEP）です。この場合、メッセージMをランダムなナンスNと連結し、ハッシュ関数hを使用してそれらを結合します。C1= M�h（N）C2 = N�h（ C1）実際には、これはラウンド関数としてhを使用する2ラウンドのFeistel暗号です。  
 次に、受信者はNをC2�h（C1）として計算し、MをC1�h（N）として回復します[212]。  
公開鍵暗号化規格にはいくつかあります。 PKCS＃1はOAEP [993]を記述しています。  
 実際、1990年代に学んだことの1つは、ランダム化により、暗号プロトコルが数学的な攻撃だけでなく、あらゆる種類の攻撃に対してより堅牢になることです。  
シグネチャを使用すると、処理が少し簡単になります。  
ただし、一部のアプリケーションでは、タイムスタンプやランダム性などの追加データを署名ブロックに含めて、サイドチャネル攻撃を困難にすることができます。  
 見事な例の1つは、Daniel BleichenbacherがSSLv 3のRSA実装を破る方法を見つけたときでした。  
cd（mod n）として復号化した場合、givencがPKCS＃1メッセージに対応するかどうかをターゲットから学習できれば、これを使用してメッセージを復号化または署名できます[264]。  
 RSAも数学的に脆弱です。準同型を使用して、RSAを破ることができます。または、同じ暗号テキストを多数の異なる小さな鍵で暗号化した場合、メッセージが短すぎる場合、または2つのメッセージが既知の多項式によって関連付けられている場合、または他のいくつかのエッジケース。  
 さらに他の攻撃には、攻撃コードをキーとして送信するか、実装が不十分な標準のパディングとして送信するかに関係なく、スタックオーバーフローが含まれています。  
7。  
7。  
 いくつかのフレーバーがあり、一部は通常のモジュラー演算を使用し、他は楕円曲線を使用します。  
pを法とする原始根は、その累乗がすべての非ゼロ数mod pを生成する数です。たとえば、モジュロ7を処理する場合、52 = 25で4（モジュロ7）に減少することがわかります。53を52⇥5または4⇥5として20に計算すると、6（モジュロ7）に減少します。 、図5のように。17 –離散対数計算の例したがって、5は7を法とする原始根です。  
 このような小さな例は検査によって解決できますが、ランダムな素数pが大きい場合、これを効率的に行う方法がわかりません。  
 つまり、これは一方向準同型です。  
5。  
2。  
 アンソニーはメッセージを受け取り、箱に入れ、南京錠をかけ、宅配便でブルータスまで運ぶことができます。  
 彼は今度は南京錠を外し、それをブルータスに持ち帰らせ、ついにそれを開けた。  
 アリスはメッセージMを受け取り、それをキーKAで暗号化して{M} KAを取得し、ボブに送信します。  
しかし、可換性のプロパティは、これが{{M} KB} KAであることを意味するため、アリスは彼女のキーKAを使用して復号化し、{M} KBを取得できます。  
適切な可換暗号化はどのように実装できますか？ワンタイムパッドは確かに通勤しますが、ここでは適していません。  
7。  
離散対数問題が助けになります。  
たとえば、アリスはhermessageをプリミティブルートgとしてエンコードし、乱数xAを選択し、gxAmodulo pを計算して、pと一緒にボブに送信します。  
 これでアリスは指数を削除します。フェルマーの定理を使用して、gxB =（gxAxB）（p�xA）（mod p）を計算し、ボブに送信します。  
 このスキームの安全性は、離散対数問題の難易度に依存します。  
5。  
2。  
 値xAおよびxBは、2つのパーティの秘密鍵にすることができます。  
 素数pとジェネレータgはすべてのユーザーに共通です。  
ボブも同じことを行い、乱数xBを選択して、yB = gxBを公開します。  
 受信すると、ボブはアリスの公開鍵yAを検索し、これもgxAxBと等しいyAxBを形成するので、メッセージを復号化できます。  
 前と同じように、素数pとジェネレーターgをすべてのユーザーに共通にしてください。  
19）。  
18 – Di-e-Hellman鍵交換プロトコルアリスとボブは、セッション鍵gRARBを使用して会話を暗号化できるようになりました。  
 このプロトコルが開始される前に対戦相手が両方のマシンを検査し、格納されている秘密鍵をすべて知っていたとしても、いくつかの基本的な条件が満たされていれば（e。  
、それらの乱数ジェネレーターは予測不可能であり、マルウェアは取り残されていなかった）セキュリティエンジニアリング190ロスアンダーソン5。  
 ASYMMETRIC CRYPTO PRIMITIVES対戦相手はまだ自分のトラフィックを盗聴することができませんでした。  
6。  
 対戦相手は、彼が取得した可能性のある以前のキーの知識から先に進むことができません。  
以下では、作業しているグループを明確に示す必要がなく、明示的に書き出す必要がない場合、RAおよびRBas DH（RA、RB）から派生したDi-e-Hellman鍵を書き込むことができます。秘密鍵RAと公開鍵gRAです。  
 パラメータpとgを選択するときは注意が必要です。たとえば、Snowdenの開示から、NSAが一般的に使用される1024ビットの素数の離散対数問題を解くことができると推測できます6。  
しかし、このプロトコルには小さな問題があります。アリスとボブは最終的にセッションキーを取得しますが、どちらもそれを誰と共有するのかという本当の考えはありません。  
 奴隷は箱をアンソニーに戻し、アンソニーは南京錠を外し、箱を開封したシーザーに戻します。  
 1つの修正は、AnthonyとBrutusがそれらのロックにシールを適用することです。  
チャーリーはアリスのボブへのメッセージを傍受し、それに返信します。同時に、彼はアリスを装ってボブと鍵交換を開始します。  
 彼がネットワークの真ん中に座ってそれらの間でメッセージを翻訳し続ける限り、彼らは彼らの通信が危険にさらされていることを検出するのに苦労するかもしれません。  
STU-2電話（現在は使用されていませんが、フォートミードのNSA博物館で見ることができます）では、2人のプリンシパルが生成したキーの8桁のハッシュを読み取り、同じ値であることを確認してから話し合いを開始します。分類された事項。  
 プロトコルは複数の攻撃、最近ではBluetooth（KNOB）攻撃のキーネゴシエーションから保護されています。これにより、中間者が簡単にブルートフォースされる1バイトのキーを強制できます。 2018年より前に製造されたすべてのデバイスは脆弱です[123]。  
 オープンレコードは795ビットで、2019年には3,100コア年を要しました[？]。NFSのバージョンを使用して、10年前よりも3倍効率的です。セキュリティエンジニアリング191ロスアンダーソン5。  
 ASYMMETRIC CRYPTO PRIMITIVEキーの長さのネゴシエーションがクリアテキストで実行されるため、攻撃者は長さを下限に強制できます。  
2。  
 Bluetoothの以前のバージョンは、セクション14で説明されているHomePlugプロトコルの「ジャストワーク」モードに似ています。  
3。  
 よりモダンなものの方が優れているように見えますが、実際には単なる劇場です。  
 公開鍵暗号化を実装している場合は、最新の標準を参照し、適切に認定されたツールキットを使用して、知識のある人に自分のしたことを評価してもらう必要があります。  
 優れた暗号化ライブラリを使用しても、足を撃つ機会がたくさんあります。  
7。  
3ElGamalデジタル署名とDSASベースpとジェネレーターgが適切な方法で選択された公開値であり、メッセージに署名したい各ユーザーが公開署名検証キーY = gXを持つ秘密署名キーXを持っていると仮定します。  
 メッセージキーkをランダムに選択し、r = gk（mod p）を形成します。  
 実行できる方程式は多数あります。 ElGamalシグネチャで偶然に使用されるものは、rX + sk = MSo sはs =（M�rX）/ kとして計算されます。これは�（p）を法として行われます。  
機能するデジタル署名スキームを取得するには、さらにいくつかの詳細を修正する必要があります。  
 また、ハッシュ関数を使用してメッセージMをハッシュし、任意の長さのメッセージに署名できるようにします。これにより、対戦相手がアルゴリズムの代数構造を使用して、署名されていないメッセージの署名を偽造できなくなります。  
7。  
DSAは、通常2048ビットの素数p7、256ビット除算の素数q（p�1）、pを法とする整数の次数qの要素g、秘密署名鍵xおよび公開検証鍵y = gxを想定しています。  
DSAは、メッセージを復元しない、ランダム化されたデジタル署名方式の典型的な例です。  
5。  
3楕円曲線暗号離散対数とその類似物は、他の多くの数学的構造に存在します。  
 これらの曲線には、それらに追加演算を定義できるプロパティがあり、結果のMordellグループは暗号化に使用できます。  
 ただし、楕円曲線暗号システムは、少なくとも2つの理由で興味深いものです。  
 楕円曲線暗号は、EMV支払いカードの最新バージョンからビットコインまでのアプリケーションで使用されています。  
 RSAとDi�e-Hellmannでは、ユーザーは自分の秘密鍵を選択し、対応する公開鍵を計算したことを思い出してください。  
 世界的な公衆がいる7 1990年代には、pは512〜1024ビット、q 160ビットの範囲になる可能性があります。これは、2001年に1023〜1024ビット[1402]に、2009年に1024〜3072ビットに変更され、qの範囲は160〜256ビット[1403]になりました。  
 ムーアの法則やアルゴリズムの進歩について心配な場合は、より大きなサイズを選択できます。  
セキュリティエンジニアリング193ロスアンダーソン5。  
 ASYMMETRIC CRYPTO PRIMITIVESkey。これを使用すると、誰でもあなたのアイデンティティへのメッセージを暗号化できます。秘密鍵を使用してこれを復号化できます。  
 どちらの場合も、あなたの秘密鍵は、自分自身だけが知っているシステム全体の秘密鍵を使用して中央当局によって計算されます。  
 メールアドレスや他の識別子からユーザーの秘密鍵を計算することはきちんとした方法のように見えるかもしれませんが、政府部門が再編成されたり名前が変更されたりすると、コストが高くなる可能性があります[115]。  
5。  
4証明機関今、公開鍵の暗号化とデジタル署名を行うことができるようになったので、ユーザーを鍵にバインドするメカニズムが必要です。  
 Loren Kohnfelderによるより一般的な解決策は、認証機関（CA）がユーザーの公開暗号鍵または署名検証鍵に署名して、ユーザーの名前、1つ以上の公開鍵、および許可などの属性を含む証明書を提供することです。  
証明書は、CA = SigKS（TS、L、A、KA、VA）（5。  
 このようにして、信頼できる方法ですべてのプリンシパルに通信する必要があるのは、管理者の公開署名検証キーだけです。  
 手始めに名前を付けるのは難しいです。分散システムに関する次の章でこれについて説明します。  
 多くの場合、政府のシステムでは、ユーザーの名前や役割だけでなく、セキュリティのクリアランスレベルも確立されます。  
 商用システムでは、多くの場合、リモートユーザーを役割ベースのアクセス制御にリンクすることです。システムエンジニアリングのレベルでの認証で失敗する可能性のある他の多くの事柄があります。  
7。  
 不良証明書の失効は、それが機能する場合、通常は不安定です。  
 これらの警告が表示されたら、最も一般的に使用されている公開鍵プロトコルであるTLSを見てみましょう。  
7。  
 これは本質的に、eコマースの開始時にTLSプロトコル（以降SSLと呼ばれます）が行うように設計されたものです。  
 ブラウザのツールバーに南京錠が表示されたときにアクティブになるプロトコルです。  
 クライアントはサーバーに、その名前C、トランザクションシリアル番号C＃、およびランダムナンスNCを含むクライアントハローメッセージを送信します。2。  
 クライアントは証明書のCSをチェックし、必要に応じて別の証明書で署名したキーをチェックし、すぐにVerisignなどの会社が発行し、ブラウザに保存したルート証明書に戻ります。3。  
 また、これまでにすべてのメッセージに対して計算されたメッセージ認証コード（MAC）を備えた終了メッセージを送信します。  
 このキーは、プレマスターシークレットキーをクライアントとサーバーで不要でないキーでハッシュすることによって計算されます。K1= h（K0、NC、NS）。  
。  
}クライアント/サーバー方向のKCSおよび{  
。  
 これらのキーは、ナンスをK1でハッシュすることによって順番に生成されます。  
 サーバーはまた、これまでのすべてのメッセージで計算されたMACを使用して、完成したメッセージを送信します。  
10できないいくつかは、ごまかしてみてください。  
 どちらの場合も、ブラウザーベンダーは迅速かつ厳しく後押ししました。Diginotarはブラックリストに登録された後で失敗しましたが、カザフの証明書は、市民が手動でインストールした場合でもブロックされました。  
セキュリティエンジニアリング195ロスアンダーソン5。  
 非対称暗号プリミティブSC！ S：C、C＃、NCS！ C：S、S＃、NS、CSC！ S：{K0} KSC！ S：{finished、MAC（K1、everythingtodate）} KCSS！ C：{finished、MAC（K1、everythingtodate）} KSC、{data} KSCOクライアントとサーバーが事前マスターシークレットを確立した後、新しいhashingitを使用してマスターシークレットを取得できるため、これ以上の公開キー操作は必要ありません。ナンス。  
7。  
1TLSの使用完全なプロトコルはこれよりも複雑であり、多数のバージョンを経てきました。  
これにより、中間者が弱い鍵の使用を強制できるダウングレード攻撃が発生しました。  
 TLSには双方向認証のオプションもあるので、クライアントが証明書も持っている場合、サーバーがこれを確認できます。  
TLSは、Webトラフィックの暗号化に使用されるだけでなく、Windows 2000以降のWindowsの認証オプションとしても使用できます。企業ネットワークでの認証にKerberosの代わりにそれを使用できます。  
5。  
5。  
0以降は健全に見えます。 SSL後のバージョン3。  
0。  
しかし、それ以来20年以上の間に、十数件の深刻な攻撃がありました。  
TLS 1。  
2は2年後、ハッシュ関数をSHA256にアップグレードし、認証済み暗号化をサポートしました。その間、出現したさまざまな攻撃を処理するパッチがいくつかありました。  
 これは、大手サービス会社がブラウザを変更して、廃止されたセキュリティエンジニアリング196ロスアンダーソン5を拒否することで対処されてきました。  
 ASYMMETRIC CRYPTO PRIMITIVESciphersuites、および厳格なトランスポートセキュリティ（STS）などの機能を追加して、awebsiteがブラウザーにhttpsを使用してのみ対話するように指示できる（ダウングレード攻撃を回避する）。  
5。  
5。  
3コアプロトコルの最新のメジャーアップグレードであるTLS 1。  
多くの古い暗号のサポートを終了するために下位互換性があり、各セッションの開始時にDi-e-Hellman鍵交換によってエンドツーエンドの転送秘密を確立することが必須になりました。  
 これはもはや不可能であるため、銀行は、廃止された暗号化を使用するという法的不快感や、エンドポイントでのコンプライアンスを監視するためにシステムを再開発するための金銭的なコストを負担する必要があります11。  
7。  
 ここでは、コード署名、PGP、QUICについて簡単に説明します。  
7。  
1コード署名コード署名は、人々がソフトウェアをディスケットではなくダウンロードするようになった1990年代に導入されました。  
 バージョンNがバージョンN + 1への更新を確認できるように、ソフトウェアに公開署名検証キーを含めることは、公開キー暗号化の単純なアプリケーションであると考えるかもしれませんが、これは事実とはかけ離れています。  
2。  
 iPhoneなどの一部のプラットフォームでは、署名されたコードのみが実行されます。これにより、ソフトウェアの出所が保証されるだけでなく、セクション22で説明するように、プラットフォーム所有者がアプリを収益化できるようになります。2;ゲーム機も似ています。  
 それが利用できない場合、検証は、マルウェア（または顧客）による改ざんを困難にするために難読化されたコードを使用して行われる可能性があります。これは、私がセクション24で説明する一定の武装競争です。  
3。  
5;商用CAに戻る信頼の連鎖があるかもしれませんが、政府による法的強制を心配する必要があります11COVID-19のパンデミックは、少し休息を与えました：マイクロソフトは2020年春にレガシーバージョンのTLSのサポートを削除する予定でしたが、これを遅らせました。  
7。  
2。  
 つまり、特にユーザーが敵である場合、コード署名は見た目ほど簡単ではありません。  
7。  
2PGP / GPGD1990年代の「暗号戦争」の最中、サイバー活動家は電子メールを暗号化する権利を求めて政府と戦い、政府は暗号化を制限する法律を求めました。歴史と政治についてはセクション26で説明します。  
7。  
S  
 GPGなどの互換性のある製品とともに、オタクの間でかなり広く使用されるようになりました。  
 また、メッセージングをサポートするために犯罪組織に販売されているカスタマイズされた電話にも組み込まれています。これについては、後でセクション25で説明します。  
1。  
 署名キーでメッセージに署名したり、目的の各受信者の公開キーを使用してメッセージを暗号化したりするためのコマンドラインオプションがあります。  
 1990年代のPGPの導入と使用から多くのことがわかりました。  
2。  
 暗号化された電子メールを他のサブジェクトに正しく送信できたのは12のサブジェクトのうち4つだけであり、すべてのサブジェクトで少なくとも1つの重大なエラーが発生しました。  
7。  
3QUICQUICは、Googleによって設計された新しいUDPベースのプロトコルであり、TLSの代替として宣伝されています。これにより、セッションの確立が迅速になり、ページの読み込み時に発生する広告オークションの待ち時間が短縮されます。セッションは、アクセスポイント間を移動する間持続できます。  
 2013年にChromeに登場し、現在インターネットトラフィックの約7％を占めています。活発な標準化コミュニティを獲得しました。  
 独立した評価によると、メリットはモバイルではなくデスクトップにある[1007]。サーバーはクライアントごとに個別の公開鍵を使用でき、これを追跡に使用できるため、プライバシーの懸念がある。  
 業界の独占傾向については、第8章で詳しく説明します。  
7。  
7。  
 これまで実際の製品に登場した2つは、しきい値暗号とブラインド署名です。  
k = nの場合、構成は簡単です。  
 。  
 + dn。  
 しきい値シグネチャは、多くのサーバーがトランザクションを個別に処理し、結果に個別に投票するシステムで最初に使用されました。最近では、「支払いは7人の取締役のうち2人が承認する必要がある」などの暗号通貨ウォレットにビジネスルールを実装するために使用されています。  
 たとえば、RSAを使用している場合、乱数Rを取り、ReM（mod n）を形成し、それを計算する署名者に提供します（ReM）d =R。  
 彼がこれを私に返すとき、Rを分割して署名M dを取得できます。  
最初はデジタルキャッシュでした。顧客に匿名の支払いトークンを発行できるようにしたい場合があります。DavidChaum氏による最初のアイデアは、シリアル番号を知らずに「デジタルコイン」に署名する方法でした[411]。  
 ブラインド署名プロトコルにより、顧客は銀行にシリアル番号を知らなくても銀行にコインを署名させることができ、プロトタイプの道路通行料システムで使用されました。  
 セクション20で説明したように、デジタルキャッシュの主な問題は、同じコインを2回使う人を検出することでしたが、これは最終的にブロックチェーンまたは他の元帳メカニズムを使用して修正されました。  
銀行も政府も支払いを匿名にすることを望んでいないため、デジタルキャッシュはうまくいきませんでした。  
 残念ながら、これはSGX（およびその同等のAMD）での証明のための設計につながりました。これは、単一の侵害されたデバイスがエコシステム全体を破壊することを意味します。  
5。  
7。  
7。  
 楕円曲線システムはこの限界を達成するように見えます。 256ビットellipticschemeは、128ビットキーを使用した128ビットブロック暗号とほぼ同じくらい解読するのが難しい場合があります。 NSAのSuite B ofmilitaryアルゴリズムで使用されている唯一の公開鍵暗号化スキームは、384ビットの楕円曲線システムです。その結果、楕円曲線暗号システムはより高速になります。  
 2019年の第3版までに、ビットコインマイナーは10分ごとに68ビットハッシュの衝突を発見しており、RSA-768は因数分解され、エドスノーデンはNSAが1024ビットプライムモジュラスのログを個別に記録できると私たちに伝えました。  
 Peter Shorは、十分に大きな量子コンピューターを構築できれば、因数分解と離散対数計算の両方が簡単になることを示しました[1725]。  
 私は（多くの物理学者がそうであるように）テクノロジーが実際のシステムに脅威を与えるかどうかについては懐疑的です。  
さらに、私はエンタングルメントベースの量子暗号が説得力のないものであることを保証するセキュリティ証明を見つけました[311]。  
 それ以来、ストリング理論[2032]のような一連のアイデアは消え去っています。  
その支持者たちは、物理学が局所的であり因果的であることが同時にありえないことを示すことになっているベルテストの謎を語ります。  
 そしてベルテストに関連する量子力学的性質を実証する古典的な力学的実験があります[1557]。  
個人的には、公開鍵暗号セキュリティエンジニアリング200ロスアンダーソン5にとって大きな課題である可能性が高いと思います。  
 ASYMMETRIC CRYPTO PRIMITIVEStographyは、楕円曲線の離散対数を計算するための優れたアルゴリズムの形で提供されます。  
量子コンピュータが機能する場合、量子コンピュータが明白な利点をもたらさない他の「ポスト量子」アルゴリズムが用意されています。  
 65回の最初の提出は、2回の審査により15に削減されました12。  
 使用中の多くのプロトコルは、Kerberosのバリアントを使用するように再設計することもできます。  
 しかし、楕円ログが簡単になると、ビットコインは偽造するのが簡単になり、暗号通貨エコシステムはおそらく崩壊し、セクション20で説明する非常に無駄なマイニング操作に終止符が打たれるでしょう。  
 したがって、惑星の未来を気にする数学者は、楕円対数問題について考えるよりも悪い結果をもたらすかもしれません。  
7。  
 確かに、20世紀に設計されたシステムに対する攻撃が実際にあり、そのほとんどは、輸出管理規則、無知な設計、またはその両方によって、あまりにも短く保たれていた鍵に関係しています。  
3。  
 後の章で、私は暗号戦争とその輸出管理規則がどのようにドアロックへの攻撃をもたらしたかの例を示します（セクション13。  
5）、携帯電話（セクション22。  
1）および著作権の執行（セクション24。  
5）。  
 第2章では、NSAバックドアを含むことが判明した楕円曲線に基づいて複雑な乱数ジェネレータを標準化するNISTのスキャンダルについて述べました。セクション2を参照してください。  
1。  
 不十分な乱数ジェネレーターは、他の多くの障害を引き起こしました：共通の要因を持つRSAキー[1140]、離散ログの予測可能なシード[1676]など。  
過去20年間に大幅な変更を余儀なくされた暗号実装への実際的な攻撃の多くは、タイミングや電力分析などのサイドチャネルを悪用しています。私はこれらに19章を捧げます。  
セキュリティエンジニアリング201ロスアンダーソン5。  
 第20章では、Signalメッセージングプロトコル、TOR匿名システム、暗号通貨など、興味深い緊急プロパティを取得するために複雑な方法で公開鍵メカニズムを使用する多くのシステムについて説明します。  
 これらには興味深い障害モードもありますが、サイドチャネルに関連するものもありますが、すべてではありません。  
5。  
 これは異なるレベルで取り組むことができます。 1つは暗号理論のレベルであり、randomoracleモデル、具象モデル、およびセマンティックセキュリティモデルについて話し合うことができ、うまくいけば、弱い動作モードやその他の構造の使用を避けることができます。  
 これらはまた、独自の数学の分野、すなわちブロック暗号解読と計算数論を持っています。  
これには、タイミング、エラー処理、電力消費、およびその他のさまざまな詳細の処理が含まれ、最新の暗号化システムが慣習を破る傾向があります。  
 どちらの場合も、かなりの数の数学があります。  
 これらには、さまざまなエラー伝播、パターン隠蔽、および整合性保護のプロパティがあります。  
 自分が何をしているかを本当に理解していない限り、ECBモードを使用しないでください。  
 特に、コンポーネントに障害が発生した場合（またはコンポーネントに障害が発生した場合）でも堅牢で、暗号化メカニズムがアクセス制御や物理的セキュリティなどの他の手段と統合されているシステムを構築することは驚くほど困難です。教訓は、自分で転がしてはいけないということです。独自のプロトコルや独自の暗号を設計しないでください。絶対に必要でない限り、独自の暗号コードを記述しないでください。  
 少なくとも、あなたの作品をピアレビューを受ける必要があります。  
8。  
研究の問題暗号研究には多くの活発なスレッドがあります。  
 ビジネスの経験的な終わりは、暗号化、署名、複合操作のためのプリミティブの設計に関係しており、これらは利用可能なプラットフォームでかなりうまく機能します。  
理論的な終わりに何が起こっているのかを知る最良の方法は、Crypto、Eurocrypt、Asiacryptなどの過去数年間の研究会議の議事録を読むことです。暗号設計に関する作業はFastSoftware暗号化に表示されます。実装への攻撃はしばしばCHESに現れます。一方、システムで暗号がどのように使用されるかについての攻撃は、IEEEのセキュリティとプライバシー、CCS、Usenixなどのトップシステムセキュリティ会議で確認できます。  
 Bruce SchneierのApplied Cryptography [1667]は、非数学者が理解できるレベルで多くの根拠をカバーしており、米国の輸出管理法にもかかわらず、1990年代に暗号コードを公開しましたが、現在は少し古いです。  
 KatzとLindellは、生徒に数学のために読んでもらう本です。  
さらに多くの専門書があります。  
Doug Stinsonの教科書には、線形クリプタナルイシス[1829]に関する別の詳細な説明があります。現代のブロック暗号理論は、Fast Software Encryptionカンファレンスシリーズの論文を通じてたどることができます。  
 Neal Koblitzは、公開鍵暗号技術[1060]の背後にある数学についての基本的な入門書を持っています。数フィールドふるいはArjenおよびHenrikLenstra [1141]によって記述されています。  
理論的な暗号学の数学的詳細に取り組みたい場合は、Dan BonehとVictor Shoup [287]による最近の大学院の教科書があります。  
8。  
 暗号学の理論的な終わりの研究は、FOCS、STOC、Crypto、Eurocrypt、およびAsiacryptの会議で見られます。  
 標準的な作業はカーン[1001]です。 Cryptologia [531、529、530]からの歴史的記事の編集、およびカーン、マークス、ウェルチマンなどによる第二次世界大戦の暗号学の歴史に関する書籍[438、1002、1224、2007]もあります。  
、一見の価値もありますが、おそらくイギリスのブレッチリーパークにある博物館が一番でしょう。  
 しかし、EllisがGCHQで働いていたため、彼の仕事は分類されたままでした。  
この話は[625]で語られています。  
そして、分類されたコミュニティはデジタル署名を事前に発明しませんでした。彼らはホイット・ディーとマーティン・ヘルマンの業績のままです。