**Parte 1**

**Primitivas: Os ingredientes da criptografia**

Bem-vindo ao mundo real da criptografia! O livro que você está segurando nas mãos (caso tenha escolhido a versão impressa) é dividido em duas partes iguais de oito capítulos. Ao percorrer todo o conteúdo, você aprenderá (quase) tudo o que há para saber sobre criptografia no mundo real — aquele em que você está inserido.

Observe que a primeira parte do livro foi escrita para ser lida em ordem, embora cada capítulo deva informar quais são os pré-requisitos, então não veja isso como uma restrição obrigatória. Os oito primeiros capítulos conduzem você pelos fundamentos — os blocos de construção da criptografia. Cada capítulo apresenta um novo ingrediente e ensina o que ele faz, como funciona e como pode ser usado com outros elementos. Esta primeira parte tem como objetivo fornecer boas abstrações e percepções antes de começarmos a usá-las na segunda parte do livro.

Boa sorte!

**Capítulo 1**

**Introdução**

Saudações, viajante; acomode-se. Você está prestes a entrar em um mundo de maravilhas e mistérios — o mundo da criptografia. Criptografia é a antiga disciplina de proteger situações atormentadas por personagens maliciosos. Este livro contém os feitiços de que precisamos para nos defender da malícia. Muitos tentaram aprender esse ofício, mas poucos sobreviveram aos desafios que impedem a maestria. Aventuras empolgantes o aguardam, de fato!

Neste livro, vamos desvendar como algoritmos criptográficos podem proteger nossas cartas, identificar nossos aliados e proteger tesouros contra nossos inimigos. Navegar pelo mar criptográfico não será a jornada mais tranquila, já que a criptografia é a base de toda a segurança e privacidade em nosso mundo — o menor erro pode ser fatal.

NOTA: Se você se sentir perdido, lembre-se de continuar avançando. Tudo fará sentido eventualmente.

**Este capítulo cobre:**

* Do que se trata a criptografia
* Criptografia teórica versus criptografia do mundo real
* O que você aprenderá ao longo desta aventura

**1.1 A criptografia trata de proteger protocolos**

Nossa jornada começa com uma introdução à criptografia, a ciência que visa defender protocolos contra sabotadores. Mas primeiro, o que é um protocolo? Simplificando, é uma lista de etapas que uma (ou mais pessoas) deve seguir para alcançar algo. Por exemplo, imagine o seguinte cenário: você quer deixar sua espada mágica desacompanhada por algumas horas para tirar uma soneca. Um protocolo para isso poderia ser:

1. Depositar a arma no chão
2. Tirar uma soneca sob uma árvore
3. Recuperar a arma do chão

Claro, não é um ótimo protocolo, já que qualquer um pode roubar sua espada enquanto você dorme... E assim, a criptografia trata de levar em conta os adversários que querem se aproveitar de você.

Nos tempos antigos, quando reis e generais estavam ocupados traindo uns aos outros e planejando golpes, um de seus maiores problemas era encontrar uma maneira de compartilhar informações confidenciais com aqueles em quem confiavam. Daí surgiu a ideia da criptografia. Foram necessários séculos e muito esforço até que a criptografia se tornasse a disciplina séria que é hoje. Agora, ela é usada em toda parte para fornecer os serviços mais básicos diante do nosso mundo caótico e adverso.

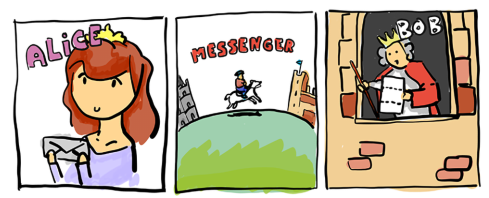
A história deste livro trata da prática da criptografia. Ele leva você a uma expedição pelo mundo da computação para abordar protocolos criptográficos em uso atualmente; também mostra do que eles são feitos e como tudo se encaixa. Enquanto um livro típico de criptografia geralmente começa com a descoberta da criptografia e percorre sua história, acredito que faz pouco sentido começar assim. Quero contar sobre o lado prático. Quero contar o que testemunhei pessoalmente, revisando aplicações criptográficas para grandes empresas como consultor, ou a criptografia que usei como engenheiro na área.

Quase não haverá fórmulas matemáticas assustadoras. O objetivo deste livro é desmistificar a criptografia, examinar o que é considerado útil hoje em dia e fornecer intuição sobre como as coisas ao seu redor são construídas. Este livro é destinado a pessoas curiosas, engenheiros interessados, desenvolvedores aventureiros e pesquisadores inquisitivos. O Capítulo 1, este capítulo, inicia um passeio pelo mundo da criptografia. Descobriremos os diferentes tipos de criptografia, quais nos interessam e como o mundo decidiu usá-los.

**1.2 Criptografia simétrica: O que é criptografia simétrica?**

Um dos conceitos fundamentais da criptografia é a criptografia simétrica. Ela é usada na maioria dos algoritmos criptográficos neste livro e, portanto, é extremamente importante. Eu introduzo esse novo conceito aqui por meio de nosso primeiro protocolo.

Vamos imaginar que a Rainha Alice precisa enviar uma carta para o Lorde Bob, que vive a alguns castelos de distância. Ela pede a seu mensageiro leal que cavalgue em seu fiel corcel e lute por terras perigosas à frente para entregar a mensagem preciosa ao Lorde Bob. No entanto, ela está desconfiada; mesmo que seu mensageiro leal a tenha servido por muitos anos, ela deseja que a mensagem em trânsito permaneça secreta para todos os observadores passivos, inclusive o mensageiro! Veja, a carta provavelmente contém alguma fofoca controversa sobre os reinos pelo caminho.



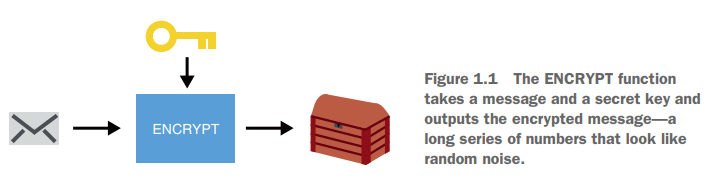
O que a Rainha Alice precisa é de um protocolo que simule entregar a mensagem ao Lorde Bob pessoalmente, sem intermediários. Este é um problema praticamente impossível de resolver na prática — a menos que introduzamos a criptografia (ou teletransporte) na equação. Isso foi o que acabamos fazendo há séculos ao inventar um novo tipo de algoritmo criptográfico — chamado de algoritmo de criptografia simétrica (também conhecido como cifra).

NOTA: Aliás, um tipo de algoritmo criptográfico é frequentemente chamado de primitiva. Você pode pensar em uma primitiva como a menor construção útil que se pode ter na criptografia, e ela é frequentemente usada com outras primitivas para construir um protocolo. É principalmente um termo, e não tem um significado particularmente importante, embora apareça com frequência suficiente na literatura para valer a pena conhecê-lo.

Vamos ver como podemos usar uma primitiva de criptografia para esconder a mensagem da Rainha Alice do mensageiro. Imagine, por ora, que a primitiva é uma caixa-preta (não podemos ver o que há dentro ou o que ela faz internamente) que fornece duas funções:

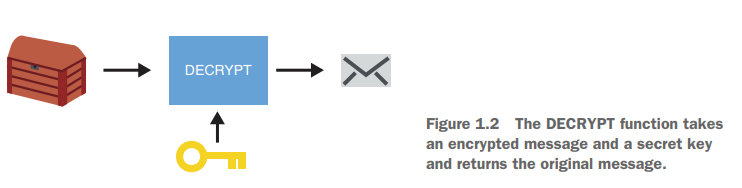
* **ENCRYPT**
* **DECRYPT**

A primeira função, **ENCRYPT**, funciona recebendo uma chave secreta (geralmente um número grande) e uma mensagem. Em seguida, ela gera uma série de números que parecem aleatórios, alguns dados ruidosos, por assim dizer. Chamaremos essa saída de **mensagem criptografada**. Eu ilustro isso na figura 1.1.



\*\*Figura 1.1\*\* - A função Encrypt recebe uma mensagem e uma chave secreta e gera a mensagem criptografada - uma longa série de números que parecem ruído aleatório

A segunda função, **DECRYPT**, é o inverso da primeira. Ela recebe a mesma chave secreta e a saída aleatória da primeira função (a mensagem criptografada) e, em seguida, recupera a mensagem original. Eu ilustro isso na figura 1.2.



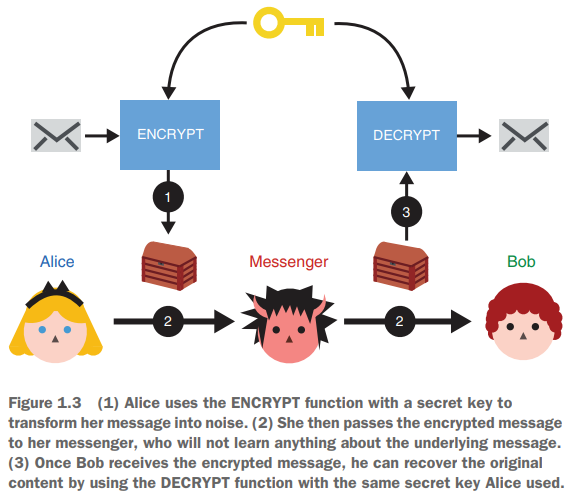
\*\*Figura 1.2\*\* - A função Decrypt pega uma mensagem criptografada e uma chave secreta e retorna a mensagem original.

Para usar essa nova primitiva, a Rainha Alice e o Lorde Bob precisam primeiro se encontrar pessoalmente e decidir qual chave secreta usar. Depois, a Rainha Alice pode usar a função **ENCRYPT** fornecida para proteger uma mensagem com a ajuda da chave secreta. Ela então passa a mensagem criptografada ao seu mensageiro, que eventualmente a entrega ao Lorde Bob. O Lorde Bob então usa a função **DECRYPT** na mensagem criptografada com a mesma chave secreta para recuperar a mensagem original. A figura 1.3 mostra esse processo.

Durante essa troca, tudo o que o mensageiro tinha era algo que parecia aleatório e que não fornecia nenhuma pista significativa sobre o conteúdo da mensagem oculta. Efetivamente, aumentamos nosso protocolo inseguro em um protocolo seguro, graças à ajuda da criptografia. O novo protocolo torna possível que a Rainha Alice entregue uma carta confidencial ao Lorde Bob sem que ninguém (exceto o Lorde Bob) saiba o conteúdo dela.

O processo de usar uma chave secreta para transformar coisas em ruído, tornando-as indistinguíveis de algo aleatório, é uma maneira comum de proteger um protocolo na criptografia. Você verá mais disso à medida que aprender mais algoritmos criptográficos nos próximos capítulos.

Aliás, a criptografia simétrica faz parte de uma categoria maior de algoritmos criptográficos chamada **criptografia simétrica** ou **criptografia de chave secreta**. Isso se deve ao fato de a mesma chave ser usada pelas diferentes funções expostas pela primitiva criptográfica. Como você verá adiante, às vezes há mais de uma chave.



\*\*Figura 1.3\*\* - (1) Alice usa a função ENCRYPT com uma chave secreta para transformar sua mensagem em ruído. (2) Ela então passa a mensagem criptografada para seu mensageiro, que não aprenderá nada sobre a mensagem subjacente. (3) Assim que Bob recebe a mensagem criptografada, ele pode recuperar o conteúdo original usando a função DECRYPT com a mesma chave secreta que Alice usou

**1.3 O princípio de Kerckhoff: Apenas a chave é mantida em segredo**

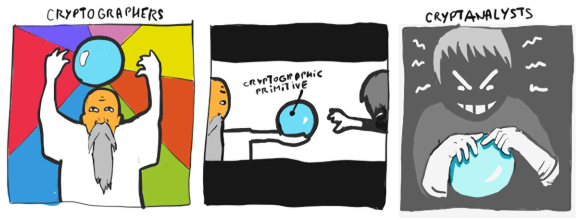
Projetar um algoritmo criptográfico (como nossa primitiva de criptografia) é uma tarefa fácil, mas projetar um algoritmo criptográfico seguro não é para os fracos de coração. Embora evitemos criar tais algoritmos neste livro, aprenderemos a reconhecer os bons. Isso pode ser difícil, pois há mais opções do que se pode pedir para essa tarefa. Pistas podem ser encontradas nos fracassos repetidos da história da criptografia, bem como nas lições que a comunidade aprendeu com eles. Ao examinarmos o passado, entenderemos o que torna um algoritmo criptográfico digno de confiança.

Centenas de anos se passaram e muitas rainhas e lordes foram enterrados. Desde então, o papel foi abandonado como nosso principal meio de comunicação em favor de tecnologias melhores e mais práticas. Hoje, temos acesso a computadores poderosos e à internet. Mais prático, claro, mas isso também significa que nosso antigo mensageiro malicioso se tornou muito mais poderoso. Ele está em todo lugar: no Wi-Fi do café Starbucks em que você está sentado, nos diferentes servidores que compõem a internet e encaminham suas mensagens, e até mesmo nas máquinas que executam nossos algoritmos. Nossos inimigos agora conseguem observar muito mais mensagens, pois cada requisição que você faz a um site pode passar pelo fio errado e ser alterada ou copiada em questão de nanossegundos sem que ninguém perceba.

Diante de nós, podemos ver que a história recente contém muitos casos de algoritmos de criptografia desmoronando, sendo quebrados por organizações estatais secretas ou por pesquisadores independentes, e falhando em proteger suas mensagens ou cumprir suas promessas. Muitas lições foram aprendidas, e aos poucos passamos a entender como produzir boa criptografia.

NOTA: Um algoritmo criptográfico pode ser considerado quebrado de várias formas. Para um algoritmo de criptografia, você pode imaginar várias formas de atacá-lo: a chave secreta pode vazar para o atacante, mensagens podem ser descriptografadas sem a ajuda da chave, alguma informação sobre a mensagem pode ser revelada apenas olhando para a mensagem criptografada, e assim por diante. Qualquer coisa que de alguma forma enfraqueça as suposições que fizemos sobre o algoritmo pode ser considerada uma quebra.

Uma noção forte surgiu do longo processo de tentativa e erro pelo qual a criptografia passou: para obter confiança nas alegações de segurança feitas por uma primitiva criptográfica, a primitiva precisa ser analisada abertamente por especialistas. Fora isso, você estará contando com **segurança pela obscuridade**, o que historicamente não funcionou bem. É por isso que os criptógrafos (as pessoas que constroem) geralmente contam com a ajuda dos criptoanalistas (as pessoas que quebram) para analisar a segurança de uma construção. (Embora os criptógrafos frequentemente também sejam criptoanalistas, e vice-versa.)



Vamos pegar como exemplo o algoritmo de criptografia AES (Advanced Encryption Standard). O AES foi produto de uma competição internacional organizada pelo Instituto Nacional de Padrões e Tecnologia dos Estados Unidos (NIST).

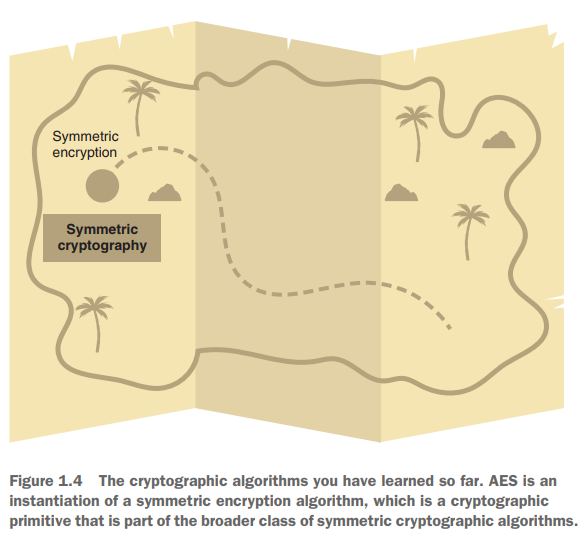
NOTA: O NIST é uma agência dos Estados Unidos cujo papel é definir padrões e desenvolver diretrizes para uso em funções governamentais, bem como em outras organizações públicas ou privadas. Assim como o AES, ele padronizou muitas primitivas criptográficas amplamente usadas.

A competição do AES durou vários anos, durante os quais muitos criptoanalistas voluntários do mundo todo se reuniram para tentar quebrar as várias construções candidatas. Após vários anos, quando confiança suficiente foi construída por meio do processo, um único algoritmo de criptografia concorrente foi nomeado para se tornar o AES.

Hoje em dia, a maioria das pessoas confia que o AES é um algoritmo de criptografia sólido, e ele é amplamente usado para criptografar quase tudo. Por exemplo, você o usa todos os dias ao navegar na internet.

A ideia de construir padrões criptográficos de forma aberta está relacionada a um conceito frequentemente chamado de **princípio de Kerckhoffs**, que pode ser entendido mais ou menos assim: seria tolice confiar que nossos inimigos não descobrirão quais algoritmos estamos usando — porque é muito provável que descubram. Em vez disso, sejamos abertos quanto a eles.

Se os inimigos da Rainha Alice e do Lorde Bob soubessem exatamente como eles estavam criptografando mensagens, como o algoritmo de criptografia deles poderia ser seguro? A resposta é: **a chave secreta!** A confidencialidade da chave é o que torna o protocolo seguro, não a confidencialidade do próprio algoritmo. Esse é um tema comum neste livro: todos os algoritmos criptográficos que aprenderemos e que são usados no mundo real são, na maioria das vezes, livres para serem estudados e usados. Apenas as chaves secretas usadas como entrada desses algoritmos são mantidas em segredo. *Ars ipsi secreta magistro* (uma arte secreta até para o mestre), disse Jean Robert du Carlet em 1644. Na próxima seção, falarei sobre um tipo totalmente diferente de primitiva criptográfica. Por ora, vamos usar a figura 1.4 para organizar o que aprendemos até agora.



\*\*Figura 1.4\*\* - Os algoritmos criptográficos que você aprendeu até agora. AES é uma instanciação de um algoritmo de criptografia simétrica, que é uma primitiva criptográfica que faz parte de uma classe mais ampla de algoritmos criptográficos simétricos.

**1.4 Criptografia assimétrica: Duas chaves são melhores que uma**

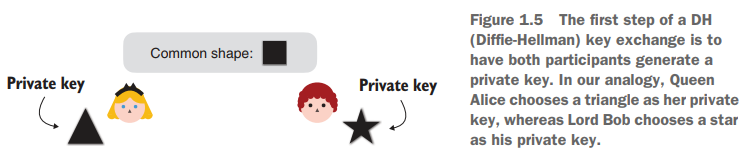
Em nossa discussão sobre criptografia simétrica, dissemos que a Rainha Alice e o Lorde Bob se encontraram primeiro para decidir uma chave simétrica. Esse é um cenário plausível, e muitos protocolos realmente funcionam assim. No entanto, isso rapidamente se torna menos prático em protocolos com muitos participantes: nosso navegador precisa se encontrar com o Google, o Facebook, a Amazon e os outros bilhões de sites antes de se conectar com segurança a eles?

Esse problema, frequentemente chamado de **distribuição de chaves**, foi difícil de resolver por muito tempo, pelo menos até a descoberta, no final da década de 1970, de outra categoria grande e útil de algoritmos criptográficos chamada **criptografia assimétrica** ou **criptografia de chave pública**. A criptografia assimétrica geralmente faz uso de chaves diferentes para funções diferentes (em oposição a uma única chave usada na criptografia simétrica) ou fornece diferentes pontos de vista para diferentes participantes. Para ilustrar o que isso significa e como a criptografia de chave pública ajuda a estabelecer confiança entre pessoas, introduzirei várias primitivas assimétricas nesta seção. Observe que isso é apenas uma amostra do que você aprenderá neste livro, pois falarei sobre cada uma dessas primitivas criptográficas em mais detalhes nos próximos capítulos.

**1.4.1 Trocas de chaves, ou como obter um segredo compartilhado**

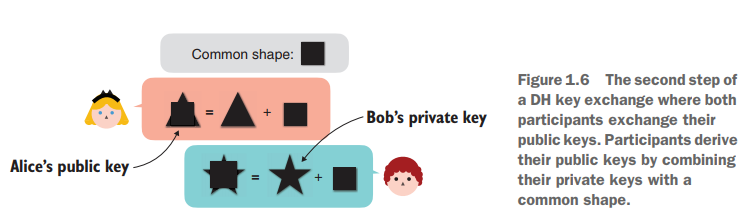
A primeira primitiva de criptografia assimétrica que veremos é a **troca de chaves**. O primeiro algoritmo de chave pública descoberto e publicado foi um algoritmo de troca de chaves nomeado em homenagem a seus autores, **Diffie-Hellman (DH)**. O principal objetivo do algoritmo de troca de chaves DH é estabelecer um segredo comum entre duas partes. Esse segredo comum pode então ser usado para diferentes propósitos (por exemplo, como chave para uma primitiva de criptografia simétrica).

No capítulo 5, explicarei como o Diffie-Hellman funciona, mas para esta introdução, vamos usar uma analogia simples para entender o que uma troca de chaves fornece. Como muitos algoritmos em criptografia, uma troca de chaves deve começar com os participantes usando um conjunto comum de parâmetros. Em nossa analogia, teremos a Rainha Alice e o Lorde Bob concordando em usar um quadrado (■). O próximo passo é que eles escolham suas próprias formas aleatórias. Ambos vão para seus respectivos locais secretos, e fora de vista, a Rainha Alice escolhe um triângulo (▲) e o Lorde Bob escolhe uma estrela (★). Os objetos escolhidos devem permanecer em segredo a todo custo! Esses objetos representam suas **chaves privadas**.



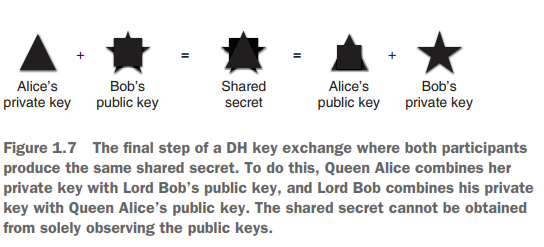
\*\*Figura 1.5\*\* - O primeiro passo da troca de chaves DH (Diffie-Hellman) é fazer com que ambos os participantes gerem uma chave privada. Em nossa analogia, a Rainha Alice escolhe um triângulo como sua chave privada, enquanto Lorde Bob escolhe uma estrela como sua chave privada.

Uma vez que escolheram suas chaves privadas, ambos combinam individualmente sua forma secreta com a forma comum que inicialmente concordaram em usar (o quadrado). As combinações resultam em formas únicas que representam suas **chaves públicas**. A Rainha Alice e o Lorde Bob agora podem trocar suas chaves públicas (daí o nome troca de chaves), pois chaves públicas são consideradas informações públicas.



\*\*Figura 1.6\*\* - A segunda etapa de uma troca de chaves DH, onde ambos os participantes trocam suas chaves públicas. Os participantes derivam suas chaves públicas combinando suas chaves privadas com um formato comum.

Agora começamos a ver por que esse algoritmo é chamado de algoritmo de chave pública. É porque ele requer um par de chaves composto por uma **chave privada** e uma **chave pública**. A etapa final do algoritmo de troca de chaves DH é bastante simples: a Rainha Alice pega a chave pública do Lorde Bob e a combina com sua chave privada. O Lorde Bob faz o mesmo com a chave pública da Rainha Alice e combina com sua própria chave privada. O resultado agora deve ser o mesmo dos dois lados; em nosso exemplo, uma forma combinando uma estrela, um quadrado e um triângulo.



\*\*Figura 1.7\*\* - A etapa final de uma troca de chaves DH, onde ambos os participantes produzem o mesmo segredo compartilhado. Para isso, a Rainha Alice combina sua chave privada com a chave pública de Lorde Bob, e Lorde Bob combina sua chave privada com a chave pública da Rainha Alice. O segredo compartilhado não pode ser obtido apenas observando as chaves públicas.

Agora cabe aos participantes do protocolo utilizar esse **segredo compartilhado**. Você verá vários exemplos disso neste livro, mas o cenário mais óbvio é utilizá-lo em um algoritmo que exige um segredo compartilhado. Por exemplo, a Rainha Alice e o Lorde Bob poderiam agora usar o segredo compartilhado como uma chave para criptografar mensagens com uma primitiva de criptografia simétrica. Para recapitular:

1. Alice e Bob trocam suas chaves públicas, que ocultam suas respectivas chaves privadas.
2. Com a chave pública do outro participante e sua respectiva chave privada, eles podem calcular um segredo compartilhado.
3. Um adversário que observa a troca de chaves públicas não tem informação suficiente para calcular o segredo compartilhado.

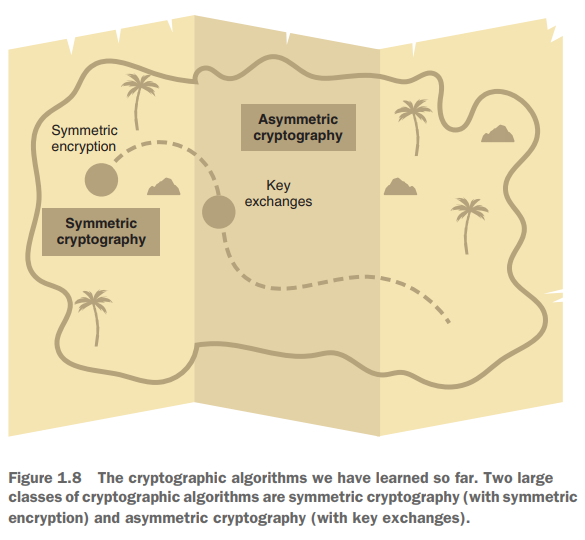
NOTA: Em nosso exemplo, o último ponto é facilmente contornável. De fato, sem o conhecimento de nenhuma chave privada, podemos combinar as chaves públicas para produzir o segredo compartilhado. Felizmente, essa é apenas uma limitação de nossa analogia, mas ela funciona bem o suficiente para entendermos o que uma troca de chaves faz.

Na prática, uma troca de chaves DH é bastante insegura. Consegue descobrir por quê?

Porque a Rainha Alice aceita qualquer chave pública que ela recebe como sendo a chave pública do Lorde Bob, eu poderia interceptar a troca e substituí-la pela minha, o que me permitiria me passar pelo Lorde Bob para a Rainha Alice (e o mesmo pode ser feito com o Lorde Bob). Dizemos que um atacante **man-in-the-middle (MITM)** pode atacar com sucesso o protocolo. Como consertamos isso? Veremos em capítulos posteriores que ou precisamos aumentar esse protocolo com outra primitiva criptográfica, ou precisamos estar cientes de antemão de qual é a chave pública do Lorde Bob. Mas então, não estamos de volta à estaca zero?

Antes, a Rainha Alice e o Lorde Bob precisavam conhecer um segredo compartilhado; agora a Rainha Alice e o Lorde Bob precisam conhecer suas respectivas chaves públicas. Como eles chegam a conhecer isso? Isso é um problema do tipo "ovo ou galinha" novamente? Bem, meio que sim. Como veremos, na prática, a criptografia de chave pública não resolve o problema da confiança, mas **simplifica seu estabelecimento** (especialmente quando o número de participantes é grande).

Vamos parar por aqui e passar para a próxima seção, pois você aprenderá mais sobre trocas de chaves no capítulo 5. Ainda temos algumas primitivas criptográficas assimétricas para descobrir para terminar nosso passeio pela criptografia do mundo real.



\*\*Figura 1.8\*\* - Os algoritmos criptográficos que aprendemos até agora. Duas grandes classes de algoritmos criptográficos são a criptografia simétrica (com criptografia simétrica) e a criptografia assimétrica (com trocas de chaves).

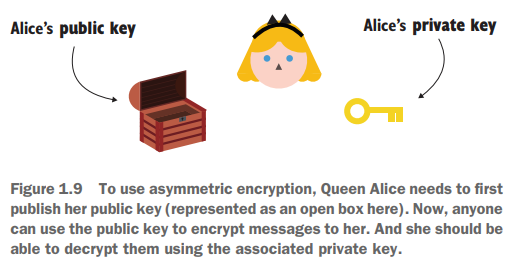
**1.4.2 Criptografia assimétrica, diferente da simétrica**

A invenção do algoritmo de troca de chaves DH foi rapidamente seguida pela invenção do algoritmo RSA, nomeado em homenagem a Ron Rivest, Adi Shamir e Leonard Adleman. O RSA contém duas primitivas diferentes: um algoritmo de criptografia de chave pública (ou **criptografia assimétrica**) e um esquema de assinatura (digital). Ambas as primitivas fazem parte da categoria maior de algoritmos criptográficos chamada de **criptografia assimétrica**. Nesta seção, explicaremos o que essas primitivas fazem e como podem ser úteis.

A primeira delas, a **criptografia assimétrica**, tem um propósito semelhante ao do algoritmo de criptografia simétrica que vimos anteriormente: ela permite criptografar mensagens para obter confidencialidade. No entanto, ao contrário da criptografia simétrica, em que os dois participantes criptografam e descriptografam mensagens com a mesma chave simétrica, a criptografia assimétrica é bem diferente:

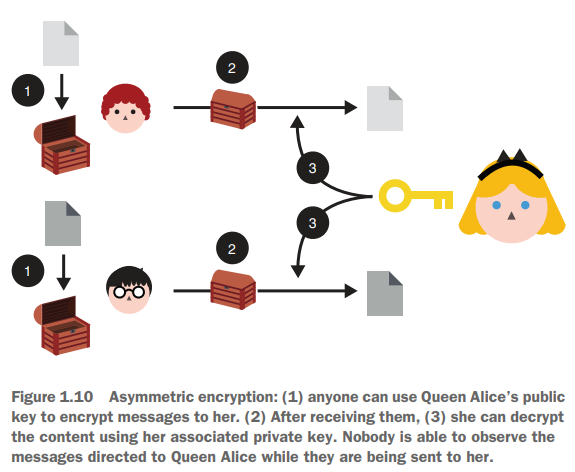
* Funciona com duas chaves diferentes: uma **chave pública** e uma **chave privada**.
* Fornece um ponto de vista assimétrico: **qualquer um pode criptografar com a chave pública**, mas **apenas o dono da chave privada pode descriptografar as mensagens**.

Vamos agora usar uma analogia simples para explicar como se pode usar a criptografia assimétrica. Começamos novamente com nossa amiga Rainha Alice, que possui uma chave privada (e sua respectiva chave pública). Vamos imaginar sua chave pública como um baú aberto que ela libera para o público para que qualquer um possa usá-lo.



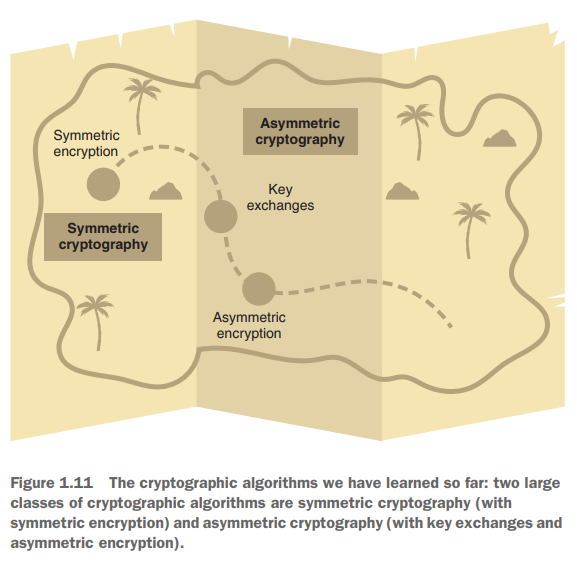
\*\*Figura 1.9\*\* - Para usar a criptografia assimétrica, a Rainha Alice precisa primeiro publicar sua chave pública (representada aqui como uma caixa aberta). Agora, qualquer pessoa pode usar a chave pública para criptografar mensagens para ela. E ela deve ser capaz de descriptografá-las usando a chave privada associada.

Agora, você e eu e qualquer um que queira pode criptografar uma mensagem para ela usando sua chave pública. Em nossa analogia, imagine que você colocaria sua mensagem dentro do baú aberto e então o fecharia. Uma vez fechado, ninguém além da Rainha Alice deveria ser capaz de abri-lo. A caixa efetivamente protege o segredo da mensagem contra observadores. A caixa fechada (ou conteúdo criptografado) pode então ser enviada para a Rainha Alice, e ela pode usar sua chave privada (conhecida apenas por ela, lembre-se) para descriptografá-la.



\*\*Figura 1.10\*\* - Criptografia assimétrica: (1) qualquer pessoa pode usar a chave pública da Rainha Alice para criptografar mensagens para ela. (2) Após recebê-las, (3) ela pode descriptografar o conteúdo usando sua chave privada associada. Ninguém consegue observar as mensagens direcionadas à Rainha Alice enquanto elas estão sendo enviadas a ela.

Vamos resumir na figura 1.11 as primitivas criptográficas que aprendemos até agora. Falta apenas mais uma para terminar nosso passeio pela criptografia do mundo real!



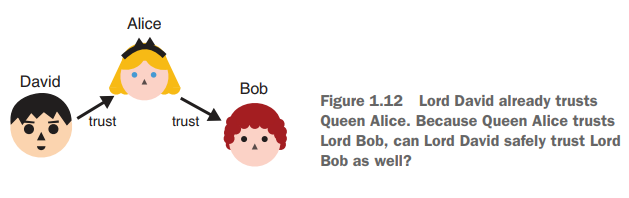
\*\*Figura 1.11\*\* - Os algoritmos criptográficos que aprendemos até agora: duas grandes classes de algoritmos criptográficos são criptografia simétrica (com criptografia simétrica e criptografia assimétrica (com trocas de chaves e criptografia assimétrica)

**1.4.3 Assinaturas digitais, como as assinaturas com caneta e papel**

Vimos que o RSA fornece um algoritmo de criptografia assimétrica, mas como mencionamos anteriormente, ele também fornece um algoritmo de **assinatura digital**. A invenção dessa primitiva criptográfica de assinatura digital tem sido de imensa ajuda para estabelecer confiança entre as Alices e Bobs do nosso mundo. É semelhante às assinaturas reais; você sabe, aquelas que você é obrigado a fazer em um contrato quando está tentando alugar um apartamento, por exemplo.

“E se falsificarem minha assinatura?”, você pode perguntar, e de fato, assinaturas reais não oferecem muita segurança no mundo real. Por outro lado, **assinaturas criptográficas** podem ser usadas da mesma forma, mas fornecem um certificado criptográfico com seu nome. Sua assinatura criptográfica **é infalsificável** e pode ser facilmente verificada por outras pessoas. Muito útil, comparado às assinaturas arcaicas que você usava para assinar cheques!

Na figura 1.12, podemos imaginar um protocolo em que a Rainha Alice quer mostrar ao Lorde David que ela confia no Lorde Bob. Este é um exemplo típico de como estabelecer confiança em um cenário com múltiplos participantes e como a criptografia assimétrica pode ajudar. Ao assinar um pedaço de papel contendo “Eu, Rainha Alice, confio no Lorde Bob”, a Rainha Alice pode se posicionar e notificar o Lorde David de que o Lorde Bob é digno de confiança. Se o Lorde David já confia na Rainha Alice e em seu algoritmo de assinatura, então ele pode escolher confiar no Lorde Bob também.



\*\*Figura 1.12\*\* - Lorde David já confia na Rainha Alice. Como a Rainha Alice confia em Lorde Bob, Lorde David pode confiar com segurança em Lorde Bob também?

Mais detalhadamente, a Rainha Alice pode usar o esquema de assinatura RSA e sua chave privada para assinar a mensagem “Eu, Rainha Alice, confio no Lorde Bob”. Isso gera uma assinatura que deve parecer um ruído aleatório.

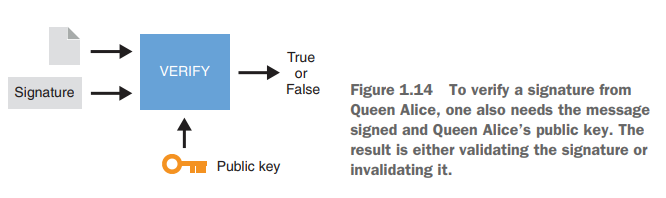


\*\*Figura 1.13\*\* - Para assinar uma mensagem, a Rainha Alice usa sua chave privada e gera uma assinatura

Qualquer um pode então verificar a assinatura combinando:

* A chave pública da Alice
* A mensagem que foi assinada
* A assinatura

O resultado será **verdadeiro** (a assinatura é válida) ou **falso** (a assinatura é inválida), conforme mostra a figura 1.14.

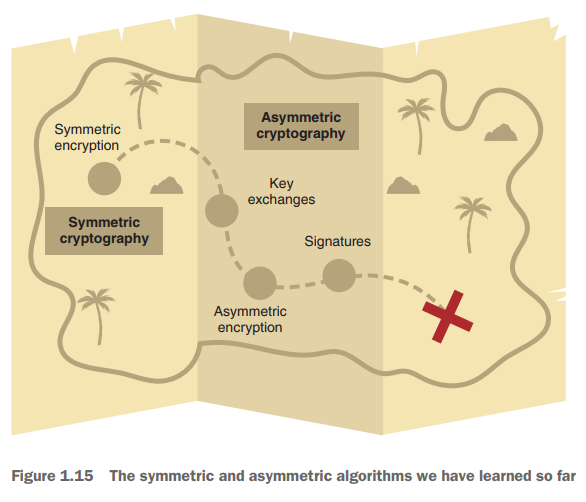


\*\*Figura 1.14\*\* - Para verificar a assinatura da Rainha Alice, também é necessária a assinatura da mensagem e a chave pública da Rainha Alice. O resultado é a validação ou invalidação da assinatura.

Agora aprendemos sobre três diferentes primitivas assimétricas:

* Troca de chaves com Diffie-Hellman
* Criptografia assimétrica
* Assinaturas digitais com RSA

Esses três algoritmos criptográficos são as primitivas mais conhecidas e comumente usadas em criptografia assimétrica. Pode não estar totalmente claro como eles podem ajudar a resolver problemas do mundo real, mas fique tranquilo, eles são usados todos os dias por muitas aplicações para proteger o que está ao redor. É hora de completar nosso panorama com todos os algoritmos criptográficos que aprendemos até agora.



\*\*Figura 1.15\*\* - Os algoritmos simétricos e assimétricos que aprendemos até agora

**1.5 Classificando e abstraindo a criptografia**

Na seção anterior, examinamos duas grandes classes de algoritmos:

* **Criptografia simétrica (ou criptografia de chave secreta)** — Um único segredo é usado. Se vários participantes conhecem o segredo, isso é chamado de segredo compartilhado.
* **Criptografia assimétrica (ou criptografia de chave pública)** — Os participantes têm uma visão assimétrica dos segredos. Por exemplo, alguns terão conhecimento de uma chave pública, enquanto outros terão conhecimento tanto da chave pública quanto da chave privada.

Criptografia simétrica e assimétrica não são as únicas duas categorias de primitivas na criptografia, e é bastante difícil classificar os diferentes subcampos. Mas, como você perceberá, grande parte do nosso livro trata (e faz uso) de primitivas simétricas e assimétricas. Isso ocorre porque grande parte do que é útil na criptografia hoje em dia está contido nesses subcampos. Outra maneira de dividir a criptografia pode ser:

* **Construções baseadas em matemática** — Dependem de problemas matemáticos como fatoração de números. (O algoritmo RSA para assinaturas digitais e criptografia assimétrica é um exemplo de tal construção.)
* **Construções baseadas em heurística** — Dependem de observações e análise estatística por criptoanalistas. (O AES para criptografia simétrica é um exemplo de tal construção.)

Há também um componente de velocidade nessa categorização, pois as construções baseadas em matemática geralmente são muito mais lentas que as construções baseadas em heurísticas. Para te dar uma ideia, construções simétricas são mais frequentemente baseadas em heurísticas (o que parece estar funcionando), enquanto a maioria das construções assimétricas é baseada em problemas matemáticos (o que se acredita ser difícil).

É difícil para nós categorizarmos rigorosamente tudo o que a criptografia tem a oferecer. De fato, todo livro ou curso sobre o assunto oferece definições e classificações diferentes. No fim das contas, essas distinções não são tão úteis para nós, pois veremos a maioria das primitivas criptográficas como **ferramentas únicas** que fazem **declarações de segurança únicas**. Podemos, por sua vez, usar muitas dessas ferramentas como blocos de construção para criar protocolos. É, portanto, essencial entender como cada uma dessas ferramentas funciona e que tipo de garantias de segurança elas oferecem para que possamos entender como elas protegem os protocolos ao nosso redor. Por esse motivo, a primeira parte deste livro abordará as primitivas criptográficas mais úteis e suas propriedades de segurança.

Muitos dos conceitos do livro podem parecer bastante complicados na primeira vez. Mas, como tudo, quanto mais lemos sobre eles e quanto mais os vemos em contexto, mais naturais eles se tornam, mais podemos **abstraí-los**. O papel deste livro é ajudá-lo a criar abstrações, permitir que você crie um **modelo mental** do que essas construções fazem e entender como elas podem ser combinadas para produzir protocolos seguros. Vou frequentemente falar sobre a **interface das construções** e dar exemplos do mundo real de uso e composição.

A definição de criptografia costumava ser simples: a Rainha Alice e o Lorde Bob querem trocar mensagens secretas. Não é mais. O que é a criptografia atualmente é algo bastante complexo de descrever e que cresceu organicamente a partir de descobertas, avanços e necessidades práticas. No fim das contas, **a criptografia é o que ajuda a aumentar um protocolo para fazê-lo funcionar em cenários adversos**.

Para entender exatamente como a criptografia pode ajudar, o conjunto de **objetivos** que esses protocolos buscam alcançar é o que importa para nós. Essa é a parte útil. A maioria das primitivas e protocolos criptográficos que aprenderemos neste livro fornecem uma ou duas das seguintes propriedades:

* **Confidencialidade** — Trata-se de ocultar e proteger informações de olhos errados. Por exemplo, a criptografia oculta as mensagens em trânsito.
* **Autenticação** — Trata-se de identificar com quem estamos falando. Por exemplo, isso pode ser útil para garantir que as mensagens que recebemos realmente venham da Rainha Alice.

Claro, isso ainda é uma grande simplificação do que a criptografia pode fornecer. Na maioria dos casos, os detalhes estão nas **declarações de segurança** das primitivas. Dependendo de como usamos uma primitiva criptográfica em um protocolo, ela alcançará diferentes propriedades de segurança.

Ao longo deste livro, aprenderemos novas primitivas criptográficas e como elas podem ser combinadas para expor propriedades de segurança como confidencialidade e autenticação. Por ora, aprecie o fato de que **a criptografia trata de fornecer garantias a um protocolo em cenários adversos**. Enquanto os “adversários” não são claramente definidos, podemos imaginar que eles são aqueles que tentam quebrar nosso protocolo: um participante, um observador, um homem no meio. Eles refletem o que um adversário da vida real poderia ser. Porque, no fim, **a criptografia é um campo prático feito para defender contra agentes mal-intencionados de carne e osso e bits**.

**1.6 Criptografia teórica vs. criptografia do mundo real**

Em 1993, Bruce Schneier lançou *Applied Cryptography* (Wiley), um livro voltado a desenvolvedores e engenheiros que desejam construir aplicações que envolvam criptografia. Por volta de 2012, Kenny Paterson e Nigel Smart iniciaram uma conferência anual chamada *Real World Crypto*, que tem como público-alvo o mesmo grupo. Mas o que significam criptografia aplicada e criptografia do mundo real? Existe mais de um tipo de criptografia?

Para responder a essas perguntas, precisamos começar definindo **criptografia teórica**, a criptografia na qual criptógrafos e criptoanalistas trabalham. Essas pessoas da cripto, na maioria das vezes, vêm da academia, trabalhando em universidades, mas às vezes da indústria ou de departamentos específicos do governo. Eles pesquisam tudo e qualquer coisa em criptografia. Os resultados são compartilhados internacionalmente por meio de publicações e apresentações em periódicos e conferências. No entanto, nem tudo que eles fazem é obviamente útil ou prático. Frequentemente, não há "prova de conceito" ou código publicado. Isso não faria sentido de qualquer forma, pois nenhum computador seria poderoso o suficiente para executar suas pesquisas. Dito isso, às vezes a criptografia teórica se torna tão útil e prática que atravessa para o outro lado.

O outro lado é o mundo da **criptografia aplicada** ou **criptografia do mundo real**. É a base da segurança que você encontra em todas as aplicações ao seu redor. Embora muitas vezes pareça não estar lá — quase transparente — ela está lá quando você acessa sua conta bancária pela internet; está com você quando envia mensagens para seus amigos; ajuda a protegê-lo quando você perde seu celular. É **onipresente** porque, infelizmente, os atacantes estão por toda parte e tentam ativamente observar e prejudicar nossos sistemas. Os profissionais geralmente vêm da indústria, mas às vezes validam algoritmos e projetam protocolos com a ajuda da comunidade acadêmica. Os resultados são frequentemente compartilhados por meio de conferências, postagens em blogs e software de código aberto.

A criptografia do mundo real geralmente se preocupa profundamente com **considerações do mundo real**:

* Qual é o nível exato de segurança oferecido por um algoritmo?
* Quanto tempo leva para executar o algoritmo?
* Qual é o tamanho das entradas e saídas exigidas pela primitiva?

A criptografia do mundo real é, como você já deve ter adivinhado, o **tema deste livro**. Embora a criptografia teórica seja o tema de outros livros, ainda daremos uma olhada no que está sendo elaborado por lá nos últimos capítulos deste livro. Esteja preparado para se surpreender, pois talvez você tenha um vislumbre da criptografia do mundo real de amanhã.

Agora você pode estar se perguntando: como os desenvolvedores e engenheiros escolhem qual criptografia usar para suas aplicações no mundo real?

**1.7 Do teórico ao prático: Escolha sua própria aventura**

No topo estão os criptoanalistas que propõem e resolvem problemas matemáticos difíceis [...]  
e na base estão os engenheiros de software que querem criptografar alguns dados.  
—Thai Duong (“So you want to roll your own crypto?”, 2020)

Em todos os anos que passei estudando e trabalhando com criptografia, nunca notei um único padrão sobre como uma primitiva criptográfica acaba sendo usada em aplicações do mundo real. As coisas são bastante caóticas. Antes que uma primitiva teórica seja adotada, há uma longa lista de pessoas que manipulam a primitiva e a moldam em algo consumível — e às vezes mais seguro — para o público em geral. Como posso explicar isso a você?

Você já ouviu falar de *Choose Your Own Adventure*? É uma antiga série de livros onde você escolhia como queria avançar pela história. O princípio era simples: você lia a primeira seção do livro; no fim da seção, o livro oferecia opções de caminhos, cada uma ligada a um número de seção para o qual você podia pular. Então, fiz o mesmo aqui! Comece lendo o próximo parágrafo e siga as instruções que ele lhe dá.

**Onde tudo começa. Quem é você?**  
Você é Alice, uma criptógrafa?  
Você é David, trabalha na iniciativa privada e precisa de uma solução para seus problemas?  
Ou é Eve, trabalha num órgão do governo e está preocupada com criptografia?

* Você é **Alice**, vá para o passo 1.
* Você é **David**, vá para o passo 2.
* Você é **Eve**, vá para o passo 3.

**Passo 1: Pesquisadores precisam pesquisar.**  
Você é um pesquisador que trabalha em uma universidade, ou na equipe de pesquisa de uma empresa privada ou organização sem fins lucrativos, ou em uma organização de pesquisa governamental como o NIST ou a NSA. Como tal, seu financiamento pode vir de diferentes lugares e pode incentivá-lo a pesquisar diferentes coisas.

* Você inventa uma nova primitiva, vá para o passo 4.
* Você inventa uma nova construção, vá para o passo 5.
* Você inicia uma competição aberta, vá para o passo 6.

**Passo 2: A indústria tem uma necessidade.**  
Como parte do seu trabalho, surge algo e você precisa de um novo padrão. Por exemplo, a *Wi-Fi Alliance* é uma organização sem fins lucrativos financiada por empresas interessadas para produzir o conjunto de padrões do protocolo Wi-Fi. Outro exemplo são os bancos que se uniram para produzir o **Payment Card Industry Data Security Standard (PCI-DSS)**, que exige o uso de certos algoritmos e protocolos para quem lida com números de cartão de crédito.

* Você decide financiar uma pesquisa necessária, vá para o passo 1.
* Você decide padronizar uma nova primitiva ou protocolo, vá para o passo 5.
* Você inicia uma competição aberta, vá para o passo 6.

**Passo 3: Um governo tem uma necessidade.**  
Você trabalha para o governo do seu país e precisa lançar uma nova criptografia. Por exemplo, o **NIST** é encarregado de publicar os **Federal Information Processing Standards (FIPS)**, que determinam quais algoritmos criptográficos podem ser usados por empresas que lidam com o governo dos EUA. Embora muitos desses padrões tenham sido casos de sucesso, e as pessoas tendam a confiar bastante em padrões propostos por órgãos governamentais, há (infelizmente) muito o que se dizer sobre falhas.

Em 2013, após revelações de Edward Snowden, descobriu-se que a **NSA** havia propositalmente — e com sucesso — promovido a inclusão de algoritmos com backdoors em padrões (veja “Dual EC: A Standardized Back Door” de Bernstein et al.), incluindo um mecanismo oculto que permitia à NSA, e somente à NSA, prever seus segredos. Esses backdoors podem ser vistos como senhas mágicas que permitem ao governo (e apenas a ele, supostamente) subverter sua criptografia. Após isso, a comunidade criptográfica perdeu muita confiança em padrões e sugestões vindos de órgãos governamentais. Recentemente, em 2019, descobriu-se que o padrão russo GOST também havia sofrido o mesmo tratamento.

Criptógrafos há muito suspeitavam de que a agência havia inserido vulnerabilidades em um padrão adotado em 2006 pelo Instituto Nacional de Padrões e Tecnologia e, depois, pela Organização Internacional de Padronização, que tem 163 países como membros.  
Memorandos confidenciais da NSA parecem confirmar que a fraqueza fatal, descoberta por dois criptógrafos da Microsoft em 2007, foi criada pela própria agência. A NSA escreveu o padrão e o promoveu agressivamente ao grupo internacional, chamando o esforço em privado de “um desafio de finesse”.  
—*New York Times*, “N.S.A. Able to Foil Basic Safeguards of Privacy on Web”, 2013

* Você financia alguma pesquisa, vá para o passo 1.
* Você organiza uma competição aberta, vá para o passo 6.
* Você pressiona pela padronização de uma primitiva ou protocolo que você já está usando, vá para o passo 7.

**Passo 4: Um novo conceito é proposto.**  
Como pesquisador, você consegue fazer o impossível; você inventa um novo conceito. Claro, alguém já pensou em criptografia, mas ainda assim novas primitivas são propostas todos os anos na criptografia. Algumas delas se mostrarão impossíveis de serem realizadas, e outras acabarão sendo viáveis. Talvez você tenha uma construção real como parte da sua proposta, ou talvez tenha que esperar para ver se alguém consegue criar algo que funcione.

* Sua primitiva é implementada, vá para o passo 5.
* Sua primitiva acaba sendo impossível de implementar, volte ao início.

**Passo 5: Uma nova construção ou protocolo é proposto.**  
Um criptógrafo ou uma equipe de criptógrafos propõe um novo algoritmo que instancie um conceito. Por exemplo, o **AES** é uma instanciação de um esquema de criptografia. (O AES foi inicialmente proposto por Vincent Rijmen e Joan Daemen, que nomearam sua construção como uma contração de seus nomes: *Rijndael*.) E agora?

* Alguém constrói algo com base na sua construção, vá para o passo 5 novamente.
* Você participa de uma competição aberta e vence! Vá para o passo 6.
* Há muito entusiasmo pelo seu trabalho; você está recebendo um padrão! Vá para o passo 7.
* Você decide patentear sua construção, vá para o passo 8.
* Você ou outra pessoa decide que será divertido implementar sua construção, vá para o passo 9.

**Passo 6: Um algoritmo vence uma competição.**  
O processo preferido dos criptógrafos é uma **competição aberta**! Por exemplo, o AES foi uma competição que convidou pesquisadores do mundo todo a participar. Após dezenas de submissões, rodadas de análise e ajuda de criptoanalistas (o que pode levar anos), a lista é reduzida a alguns candidatos (no caso do AES, um único), que então é promovido a padrão.

* Você teve sorte: após muitos anos de competição, sua construção venceu! Vá para o passo 7.
* Infelizmente, você perdeu. Volte ao início.

**Passo 7: Um algoritmo ou protocolo é padronizado.**  
Um padrão geralmente é publicado por um governo ou por um órgão de padronização. O objetivo é garantir que todos estejam na mesma página para **maximizar a interoperabilidade**. Por exemplo, o **NIST** publica padrões criptográficos regularmente. Um conhecido órgão de padronização em criptografia é a **Internet Engineering Task Force (IETF)**, que está por trás de muitos padrões da internet (como TCP, UDP, TLS etc.) e que você ouvirá bastante neste livro. Os padrões no IETF são chamados de **Request For Comments (RFC)** e podem ser escritos por praticamente qualquer pessoa que deseje escrever um padrão.

Para reforçar que não votamos, também adotamos a tradição do “zumbido”:  
Quando, por exemplo, temos reuniões presenciais e o coordenador do grupo de trabalho quer ter uma “noção do ambiente”, ao invés de levantar as mãos, às vezes o coordenador pede que cada lado emita um zumbido sobre determinada questão, seja “a favor” ou “contra”.  
—RFC 7282 (“On Consensus and Humming in the IETF”, 2014)

Às vezes, uma empresa publica um padrão diretamente. Por exemplo, a **RSA Security LLC** (financiada pelos criadores do algoritmo RSA) lançou uma série de 15 documentos chamados **Public Key Cryptography Standards (PKCS)** para legitimar algoritmos e técnicas que a empresa usava na época. Hoje em dia, isso é bem raro, e muitas empresas passam pelo IETF para padronizar seus protocolos ou algoritmos como um RFC, em vez de um documento proprietário.

* Seu algoritmo ou protocolo é implementado, vá para o passo 9.
* Ninguém se importa com seu padrão, volte ao início.

**Passo 8: Uma patente expira.**  
Uma patente em criptografia geralmente significa que ninguém vai usar o algoritmo. Uma vez que a patente expira, não é incomum ver um **renovado interesse** pela primitiva. O exemplo mais famoso provavelmente são as **assinaturas Schnorr**, que foram as primeiras candidatas a se tornar o esquema de assinatura mais popular até que Schnorr ele mesmo patenteou o algoritmo em 1989. Isso levou o NIST a padronizar um algoritmo inferior chamado **DSA (Digital Signature Algorithm)**, que se tornou o esquema de assinatura mais comum na época, mas que hoje em dia tem pouco uso. A patente das assinaturas Schnorr expirou em 2008, e o algoritmo desde então tem retomado popularidade.

* Demorou demais, seu algoritmo será esquecido para sempre. Volte ao início.
* Sua construção inspira muitas outras a serem inventadas com base nela, vá para o passo 5.
* Agora as pessoas querem usar sua construção, mas não antes de ela ser padronizada de verdade. Vá para o passo 7.
* Alguns desenvolvedores estão implementando seu algoritmo! Vá para o passo 9.

**Passo 9: Uma construção ou protocolo é implementado.**  
Os implementadores têm a tarefa difícil de não apenas decifrar um artigo ou um padrão (embora padrões sejam supostamente voltados aos implementadores), mas também de tornar suas implementações **fáceis e seguras de usar**. Isso nem sempre é uma tarefa simples, já que muitos bugs devastadores podem surgir na forma como a criptografia é utilizada.

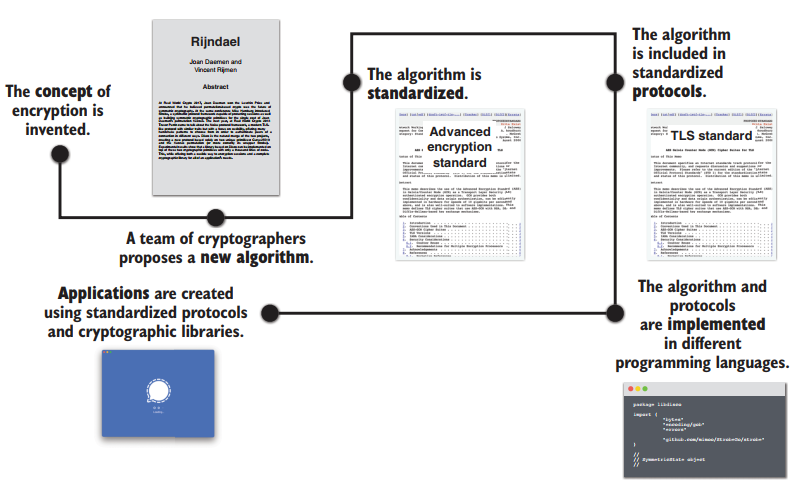
* Alguém decide que está na hora dessas implementações serem respaldadas por um padrão. Sem isso, é constrangedor. Vá para o passo 7.
* Está chovendo hype sobre sua biblioteca criptográfica! Vá para o passo 10.

**Passo 10: Um desenvolvedor usa um protocolo ou primitiva em uma aplicação.**  
Um desenvolvedor tem uma necessidade, e sua biblioteca criptográfica parece resolvê-la — simples assim!

* A primitiva resolve o problema, mas não tem um padrão. Não é ótimo. Vá para o passo 7.
* Gostaria que isso fosse escrito na minha linguagem de programação. Vá para o passo 9.
* Eu usei a biblioteca errado ou a construção está quebrada. **Game over.**

Você entendeu! Há muitas maneiras para uma primitiva chegar ao mundo real.  
A melhor forma envolve **muitos anos de análise**, **um padrão amigável ao implementador** e **boas bibliotecas**.  
Uma forma pior envolve **um algoritmo ruim com uma implementação fraca**.

Na figura 1.16, eu ilustro o caminho preferido:



**1.8 Uma palavra de advertência**

Qualquer um, desde o amador mais desinformado até o melhor criptógrafo,  
pode criar um algoritmo que ele mesmo não consegue quebrar.  
—Bruce Schneier (“Memo to the Amateur Cipher Designer”, 1998)

Preciso adverti-lo: a arte da criptografia é difícil de dominar. Seria imprudente supor que você pode construir protocolos criptográficos complexos assim que terminar este livro. Esta jornada deve iluminá-lo, mostrar o que é possível e como as coisas funcionam, mas **não fará de você um mestre da criptografia**.

Este livro **não é o santo graal**. De fato, as últimas páginas deste livro conduzem você à lição mais importante — **não vá sozinho em uma aventura real**. Dragões podem matar, e você precisa de apoio ao seu lado para derrotá-los. Em outras palavras: a criptografia é complicada, e este livro, sozinho, não o autoriza a abusar do que aprendeu. Para construir sistemas complexos, **são necessários especialistas** que estudaram seu ofício durante anos. Em vez disso, o que você aprenderá aqui é a reconhecer **quando a criptografia deve ser usada**, ou, se algo parecer suspeito, **quais primitivas e protocolos criptográficos estão disponíveis** para resolver os problemas que você enfrenta, e **como todos esses algoritmos funcionam por baixo dos panos**.

Agora que você foi alertado, siga para o próximo capítulo.

**Resumo**

* Um **protocolo** é uma receita passo a passo onde vários participantes tentam alcançar algo como a troca de mensagens confidenciais.
* A **criptografia** trata de aumentar os protocolos para torná-los seguros em contextos adversos. Isso geralmente requer segredos.
* Uma **primitiva criptográfica** é um tipo de algoritmo criptográfico. Por exemplo, criptografia simétrica é uma primitiva criptográfica, enquanto o AES é um algoritmo específico de criptografia simétrica.
* Uma forma de classificar as diferentes primitivas criptográficas é dividi-las em dois tipos: **criptografia simétrica** e **criptografia assimétrica**.
  + A criptografia simétrica usa uma única chave (como visto com a criptografia simétrica),
  + Enquanto a criptografia assimétrica faz uso de chaves diferentes (como visto com trocas de chaves, criptografia assimétrica e assinaturas digitais).
* As **propriedades criptográficas** são difíceis de classificar, mas geralmente visam fornecer uma destas duas propriedades:
  + **Autenticação**, que verifica a autenticidade de algo ou alguém.
  + **Confidencialidade**, que garante a privacidade dos dados ou identidades.
* A **criptografia do mundo real** importa porque é onipresente em aplicações tecnológicas, enquanto a **criptografia teórica** é muitas vezes menos útil na prática.
* A maioria das primitivas criptográficas contidas neste livro foi acordada após **longos processos de padronização**.
* A criptografia é **complicada**, e há muitos perigos ao implementar ou usar primitivas criptográficas.