WhatsApp

Visão Geral da Criptografia do WhatsApp

Documento Técnico

Versão 3 atualizada em 22 de outubro de 2020
Versão 2 atualizada em 19 de dezembro de 2017
Versão 1 original publicada em 5 de abril de 2016

Índice

Apresentação

Este documento oferece uma explicação técnica do sistema de criptografia de ponta a ponta do WhatsApp. Para mais informações, acesse o site do WhatsApp: www.whatsapp.com/security

Com o WhatsApp Messenger, os usuários podem trocar mensagens (incluindo conversas individuais, conversas em grupo, imagens, vídeos, mensagens de voz e arquivos), compartilhar atualizações de status e fazer chamadas por meio do WhatsApp em todo o mundo. As mensagens e chamadas de voz e de vídeo entre um remetente e um destinatário que usam o software cliente do WhatsApp lançado depois do dia 31 de março de 2016 utilizam o Protocolo Signal descrito abaixo. Consulte "Definição de criptografia de ponta a ponta" para informações sobre quais comunicações são protegidas com a criptografia de ponta a ponta.

O Protocolo Signal, desenvolvido pela Open Whisper Systems, é a base para a criptografia de ponta a ponta do WhatsApp. Esse protocolo de criptografia de ponta a ponta foi desenvolvido para impedir que terceiros e o WhatsApp tenham acesso a chamadas e mensagens em texto não criptografado. Além disso, mesmo que as chaves de criptografia do dispositivo de um usuário estejam fisicamente comprometidas, elas não podem ser usadas para voltar no tempo e descriptografar mensagens transmitidas anteriormente.

Este documento fornece uma visão geral do Protocolo Signal e seu uso no WhatsApp.

Termos

Tipos de chave pública

- `Par de chaves de Identidade` – Um par de chaves Curve25519 de longo prazo, gerado no momento da instalação.
- `Pré-chave Assinada` – Um par de chaves Curve25519 de médio prazo, gerado no momento da instalação, assinado pela `Chave de Identidade`, e alternada periodicamente.
- `Pré-chaves de Uso Único` – Uma fila de pares de chaves Curve25519 para uso único, gerados no momento da instalação e repostos conforme necessário.

Tipos de chaves de sessão

- `Chave Raiz` – Um valor de 32 bytes que é usado para criar `Chaves Corrente`.
- `Chave Corrente` – Um valor de 32 bytes que é usado para criar `Chaves de Mensagem`.
- `Chave de Mensagem` – Um valor de 80 bytes que é usado para criptografar o conteúdo das mensagens. 32 bytes são usados para uma chave AES-256, 32 bytes para uma chave HMAC-SHA256 e 16 bytes para uma chave IV.

Registro do cliente

No momento do registro, um cliente do WhatsApp transmite a `Chave de Identidade` pública, a `Pré-chave Assinada` pública (com sua assinatura) e um lote de `Pré-chaves de Uso Único` públicas para o servidor. O servidor do WhatsApp armazena essas chaves públicas associadas ao identificador do usuário.

Configuração da sessão inicial

Para se comunicar com outro usuário do WhatsApp, um cliente do WhatsApp precisa primeiro estabelecer uma sessão criptografada. Uma vez estabelecida a sessão, os clientes não precisam reconstruir uma nova sessão entre si até que o estado da sessão existente seja perdido por razão de um evento externo, como a reinstalação do app ou a troca de aparelho.

Para estabelecer uma sessão:

1. O cliente que inicia a sessão (“iniciador”) solicita a `Chave de Identidade` pública, a `Pré-chave Assinada` pública e uma `Pré-chave de Uso Único` pública para o destinatário.
2. O servidor retorna os valores da chave pública solicitados. A `Pré-chave de Uso Único` é usada somente uma vez e é removida do armazenamento do servidor após ter sido solicitada. Se o último lote do destinatário das `Pré-chaves de Uso Único` for usado e o destinatário não fizer a reposição, nenhuma `Pré-chave de Uso Único` será retornada.
3. O iniciador salva a `Chave de Identidade` do destinatário como `destinatário I`, a `Pré-chave Assinada` como destinatário `S`, e a `Pré-chave de Uso Único` como destinatário `O`.
4. O iniciador gera um par de chave Curve25519 efêmero, iniciador `E`.
5. O iniciador carrega a própria `Chave de Identidade` como

iniciador `I`.

6. O iniciador calcula um segredo mestre como `master_secret = ECDH (iniciador I, destinatário S) || ECDH (iniciador E, destinatário I) || ECDH (iniciador E, destinatário S) || ECDH (iniciador E, destinatário O)`.
   Se não houver uma `Pré-chave de Uso Único`, o ECDH final será omitido.
7. O iniciador usa HKDF para criar uma `Chave Raiz` e `Chaves Corrente` do `master_secret`.

Configuração da sessão do destinatário

Depois de construir uma sessão de criptografia de longa duração, o iniciador pode começar a enviar mensagens ao destinatário imediatamente, mesmo que o destinatário esteja offline. Até o destinatário responder, o iniciador inclui as informações (no cabeçalho de todas as mensagens enviadas) que o destinatário precisa para criar uma sessão correspondente. Isso inclui o iniciador `iniciador E` e o `iniciador I` do iniciador.

Quando o destinatário recebe uma mensagem que inclui informações sobre a configuração da sessão:

1. O destinatário calcula o `master_secret` correspondente usando as próprias chaves privadas e as chaves públicas anunciadas no cabeçalho da mensagem recebida.
2. O destinatário exclui a `Pré-chave de Uso Único` utilizada pelo iniciador.
3. O iniciador usa HKDF para derivar uma `Chave Raiz` e `Chaves Corrente` correspondentes do `master_secret`.

Troca de mensagens

Uma vez estabelecida a sessão, os clientes trocam mensagens protegidas com uma `Chave de Mensagem` usando `AES256` no modo CBC para criptografia e `HMAC-SHA256` para autenticação.

A `Chave de Mensagem` muda para cada mensagem transmitida, e é efêmera, de tal forma que a `Chave de Mensagem` usada para criptografar uma mensagem não pode ser reconstruída a partir do estado da sessão depois de uma mensagem ter sido transmitida ou recebida.

A `Chave de Mensagem` é derivada de uma `Chave Corrente` do remetente que é "ajustada" e encaminhada com todas as mensagens enviadas. Além disso, um novo acordo ECDH é realizado com todas as mensagens que vão e voltam para criar uma nova `Chave Corrente`. Isto faz com que haja sigilo pela combinação do "hash ratchet" imediato e uma "DH ratchet".

Cálculo de uma `Chave de Mensagem` a partir de uma `Chave Corrente`

Toda vez que uma nova `Chave de Mensagem` é necessária para o remetente da mensagem, o seguinte cálculo é feito:

1. `Chave de Mensagem = HMAC-SHA256(Chave Corrente, 0x01)`.
2. Assim, a `Chave Corrente` é atualizada como `Chave Corrente = HMAC-SHA256(Chave Corrente, 0x02)`.

Isso faz com que a Chave Corrente seja "ajustada" e encaminhada. E também significa que uma `Chave de Mensagem` armazenada não pode ser usada para derivar valores atuais ou passados da `Chave Corrente`.

Cálculo de uma `Chave Corrente` a partir de uma `Chave Raiz`

Toda vez que uma mensagem é transmitida, uma chave pública `Curve25519` efêmera é anunciada em conjunto. Quando uma resposta é recebida, uma nova `Chave Corrente` e uma `Chave Raiz` são calculadas da seguinte forma:

1. `ephemeral_secret = ECDH(remetente efêmero, destinatário efêmero)`.
2. `Chave Corrente, Chave Raiz = HKDF(Chave Raiz, ephemeral_secret)`.

Uma chave corrente é apenas usada para enviar as mensagens de um usuário, portanto, as chaves de mensagem não são reutilizadas. Devido à forma que as `Chaves de Mensagem` e as `Chaves Corrente` são calculadas, as mensagens podem chegar atrasadas, fora de ordem, ou podem ser totalmente perdidas sem nenhum problema.

Transmissão de arquivos de mídia e outros anexos

Anexos de qualquer tipo (vídeo, áudio, imagens ou arquivos) também são protegidos com criptografia de ponta a ponta:

1. Quando o usuário do WhatsApp envia uma mensagem (“remetente”), são geradas uma chave efêmera de 32 bytes `AES256` e uma chave efêmera de 32 bytes `HMAC-SHA256`.
2. O remetente criptografa o anexo com a chave `AES256` no modo CBC com um IV aleatório e, em seguida, anexa um MAC do texto codificado usando HMAC-SHA256.
3. O remetente carrega o anexo criptografado para um armazenamento de blob (*blob store*).
4. O remetente transmite ao destinatário uma mensagem criptografada normal que contém a chave de criptografia, a chave HMAC, um hash SHA256 do blob criptografado, e um ponteiro para o blob no armazenamento de blob.
5. O destinatário descriptografa a mensagem, recupera o blob criptografado de armazenamento de blob, confirma o hash SHA256, verifica o MAC e lê o texto não criptografado.

Mensagens em grupo

Os apps tradicionais de mensagens não criptografadas normalmente empregam "fan-out do lado do servidor" para mensagens em grupo. Um cliente que envia uma mensagem para um grupo de usuários transmite uma única mensagem, que é distribuída N vezes para os N diferentes membros do grupo pelo servidor.

Isso é o oposto de "*fan-out* do lado do cliente", em que um cliente transmitiria uma única mensagem N vezes para os próprios N diferentes membros do grupo.

As mensagens para os grupos do WhatsApp são baseadas nas sessões criptografadas em pares descritas acima para obter um fan-out eficiente do lado do servidor para a maioria das mensagens enviadas a grupos. Isso é feito usando o componente "Chaves de Remetente" do Protocolo de Mensagens Signal.

Na primeira vez que um membro de um grupo do WhatsApp envia uma mensagem para um grupo:

1. O remetente gera uma `Chave Corrente` aleatória de 32 bytes.
2. O remetente gera um par de chaves de `Chave de Assinatura` Curve25519 aleatória.
3. O remetente combina a `Chave Corrente` de 32 bytes e a chave pública da `Chave de Assinatura` em uma mensagem de `Chave de Remetente`.
4. O remetente criptografa individualmente a `Chave de Remetente` para cada membro do grupo, usando o protocolo de mensagens em pares conforme explicado anteriormente.

Para todas as mensagens subsequentes para o grupo:

1. O remetente deriva uma `Chave de Mensagem` da `Chave Corrente` e atualiza a `Chave Corrente`.
2. O remetente criptografa a mensagem usando `AES256` no modo CBC.
3. O remetente assina o texto codificado usando a `Chave de Assinatura`.
4. O remetente transmite a única mensagem do texto codificado ao servidor, que faz *fan-out* do lado do servidor para todos os participantes do grupo.

O “hash ratchet” da `Chave Corrente` do remetente da mensagem fornece sigilo no encaminhamento. Sempre que um membro sai do grupo, todos os participantes limpam a `Chave de Remetente` e começam de novo.

Configuração da chamada

As chamadas de voz e de vídeo também são protegidas com a criptografia de ponta a ponta. Quando um usuário do WhatsApp inicia uma chamada de voz ou de vídeo:

1. Caso já não exista uma, o iniciador cria a sessão criptografada com o destinatário (como descrito na seção *Configuração da Sessão Inicial*).
2. O iniciador gera um segredo mestre SRTP aleatório de 32 bytes.
3. O iniciador transmite uma mensagem criptografada ao destinatário que indica uma chamada recebida, e contém o segredo mestre SRTP.
4. Se o respondedor atender a chamada, será feita uma chamada criptografada SRTP.

Status

Os status do WhatsApp são criptografados da mesma forma que as mensagens em grupo. A primeira atualização de status enviada a um determinado conjunto de destinatários segue a mesma sequência de etapas que a primeira vez em que um membro do grupo do WhatsApp envia uma mensagem para um grupo. Do mesmo modo, as atualizações de status subsequentes enviadas ao mesmo conjunto de destinatários seguem a mesma sequência de passos que todas as mensagens subsequentes para um grupo. Quando um remetente da atualização de status remove um destinatário alterando as configurações de privacidade de status ou removendo o número da lista de contatos, o remetente de status limpa a Chave de Remetente e começa de novo.

Localização em tempo real

As mensagens de localização em tempo real e as atualizações são criptografadas da mesma forma que as mensagens em grupo. A primeira localização em tempo real ou atualização enviada segue a mesma sequência de etapas que a primeira vez em que um membro do grupo do WhatsApp envia uma mensagem no grupo. Entretanto, as localizações em tempo real exigem um alto volume de transmissões e atualizações de localização com perdas na entrega, em que os destinatários podem esperar grandes saltos no número de ajustes, ou contagens de iteração. O Protocolo Signal utiliza um algoritmo de tempo linear para ajustar (ratcheting) o que for muito lento para esta aplicação. Este documento oferece um algoritmo rápido de ajuste (ratcheting) para resolver este problema.

No momento, as Chaves Corrente são unidimensionais. Para ajustar `N` etapas, é preciso `N` cálculos. As Chaves Corrente são indicadas como `CK (contagem de iteração)` e Chaves de Mensagem como `MK (contagem de iteração)`.

```
CK(0)
 ↓
CK(1)
 ↓
 …
 ↓
CK(N-1) → MK(N-1)
```

Considere uma extensão onde guardamos duas séries de chaves corrente:

```
CK₁(0) → CK₂(0)
 ↓          ↓
CK₁(1)    CK₂(1)
            ↓
            …
            ↓
          CK₂(M-1) → MK(M-1)
```

Neste exemplo, as Chaves de Mensagem são sempre derivadas de CK_2. Um destinatário que precisa de muitos ajustes pode pular `M` iterações de cada vez (em que `M` é um inteiro positivo constante estabelecido) pelo ajuste (ratcheting) do CK_1 e gera um novo CK_2:

```
CK₁(0)
 ↓
CK₁(1) → CK₂(0) → MK(M)
 ↓          ↓
CK₁(2)    CK₂(1)
```

Um valor de CK_2 pode ser aumentado para `M` vezes. Para ajustar `N` etapas, é preciso cálculos de até `[N÷M] + M`.

Depois que um remetente cria uma Chave de Mensagem e a utiliza para criptografar uma mensagem, todas as Chaves Corrente no caminho que levou à criação devem ser destruídas para preservar o sigilo do encaminhamento.

Generalizando para as dimensões `D`, um remetente pode produzir Chaves Corrente iniciais `D`. Todas as chaves corrente, exceto a primeira, são derivadas da chave corrente anterior usando uma função unidirecional distinta: estas são as setas que apontam para a direita no diagrama acima. Os remetentes distribuem todas as chaves corrente `D` para os destinatários que precisam delas, exceto como observado abaixo.

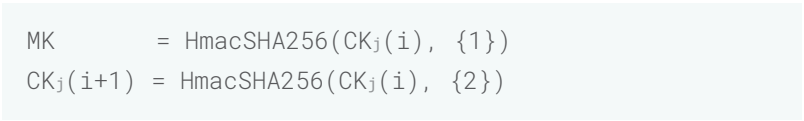
Os valores legais para `D` são potências positivas de dois menores ou iguais ao número de bits na contagem de iteração: 1, 2, 4, 8, 16, e 32. Os implementadores selecionam um valor de `D` como uma troca de memória explícita da CPU (ou largura de banda da rede da CPU).

Se uma chave corrente CK_j (para `j` em `[1, D]`) tiver uma contagem de iteração de `M`, ela não poderá ser usada. Este algoritmo restaura as chaves corrente para um estado utilizável:

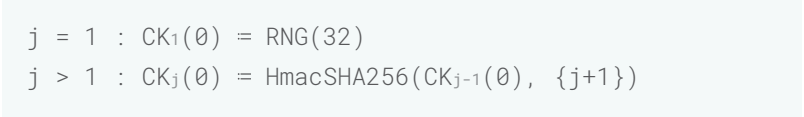
1. Se `j` = 1, há uma falha porque a contagem de iteração atingiu o limite.
2. Derive CK_j de CK_{j-1}
3. Ajuste CK_{j-1} uma vez, repetindo se necessário.

Passar de uma contagem de iteração para outra nunca ajusta uma única chave corrente mais do que M vezes. Portanto, nenhuma operação de ajuste (ratcheting) leva mais que $D \times M$ etapas.

O Signal utiliza funções diferentes para o cálculo do ajuste (ratcheting) com a chave de mensagem, uma vez que ambos são provenientes da mesma chave corrente. Nesta notação, `{x}` refere-se a uma matriz de bytes contendo um único byte `x`.



Cada dimensão deve usar uma função diferente. As chaves são inicializadas como:



E ajustadas como:

$$CK_j(i) \coloneqq HmacSHA256(CK_j(i-1), \{j+1\})$$

Verificando as chaves

Os usuários do WhatsApp também têm a opção de verificar as chaves de outros usuários que estão se comunicando dentro de um contexto de criptografia de ponta a ponta para que possam confirmar que um terceiro não autorizado (ou o WhatsApp) não começou um ataque "*man-in-the-middle*" (um ataque na segurança que pode interceptar ou modificar os dados transmitidos entre os usuários). Isso pode ser feito ao ler um código QR, ou ao comparar um número de 60 dígitos.

O código QR contém:

1. Uma versão.
2. O identificador de usuário para ambas as partes.
3. A `Chave de Identidade` pública de 32 bytes para ambas as partes.

Quando um usuário lê o código QR do outro, as chaves são comparadas para garantir que o que está no código QR corresponda à `Chave de Identidade` conforme recuperada do servidor.

O número de 60 dígitos é calculado concatenando as duas impressões digitais numéricas de 30 dígitos para a `Chave de Identidade` de cada usuário. Para calcular uma impressão digital numérica de 30 dígitos:

1. O SHA-512 combina iterativamente a `Chave de Identidade` pública e o identificador de usuário 5200 vezes.
2. Pegue os primeiros 30 bytes do hash final de saída.
3. Divida o resultado de 30 bytes em seis partes de 5 bytes.
4. Converta cada parte de 5 bytes em 5 dígitos interpretando cada uma como um *big-endian* inteiro sem assinatura e reduzindo-a no módulo 100000.
5. Concatene os seis grupos de cinco dígitos em trinta dígitos.

Segurança do transporte

Toda comunicação entre os clientes e os servidores do WhatsApp é feita em um canal criptografado separado. Em dispositivos Windows Phone, iPhone e Android, clientes com capacidade de criptografia de ponta a ponta usam "noise pipes" com Curve25519, AES-GCM e SHA256 do Noise Protocol Framework para conexões interativas de longa duração.

Isso proporciona aos clientes algumas boas propriedades:

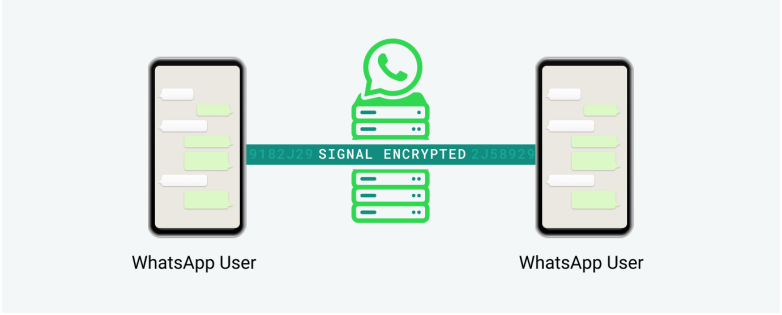
1. Configuração e retomada de conexões extremamente rápidas e leves.
2. Criptografa os metadados para escondê-los dos observadores não autorizados da rede. Nenhuma informação sobre a identidade do usuário de conexão é divulgada.
3. Nenhum segredo de autenticação do cliente é armazenado no servidor. Os clientes se autenticam usando um par de chaves Curve25519. Assim, o servidor armazena apenas a chave de autenticação pública do cliente. Se o banco de dados de usuários do servidor for comprometido, nenhuma credencial de autenticação privada será divulgada.

Nota: Nos casos em que um usuário do WhatsApp Business delega a operação da sua Business API a um prestador de serviço, esse prestador de serviço terá acesso às suas chaves privadas - mesmo que esse prestador de serviço seja o Facebook. No entanto, essas chaves privadas ainda não serão armazenadas no servidor de conversas do WhatsApp. Veja abaixo os detalhes.

Definição da criptografia de ponta a ponta

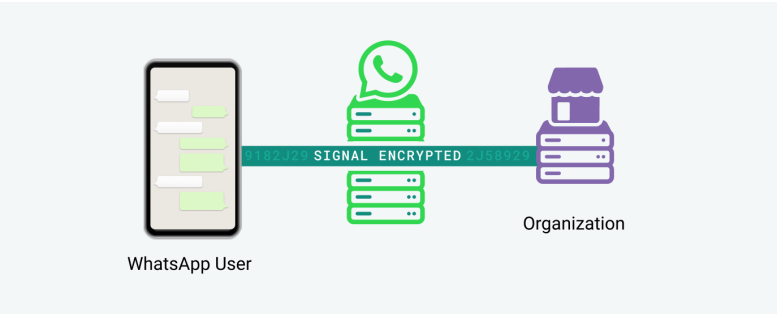
O WhatsApp define criptografia de ponta a ponta como comunicações que permanecem criptografadas em um dispositivo controlado pelo remetente para um dispositivo controlado pelo destinatário do qual terceiros não podem acessar esse conteúdo, nem mesmo o WhatsApp ou a empresa controladora Facebook. Um terceiro nesse contexto significa qualquer organização que não seja o remetente ou destinatário que participa diretamente da conversa

Implementação dos Serviços do WhatsApp

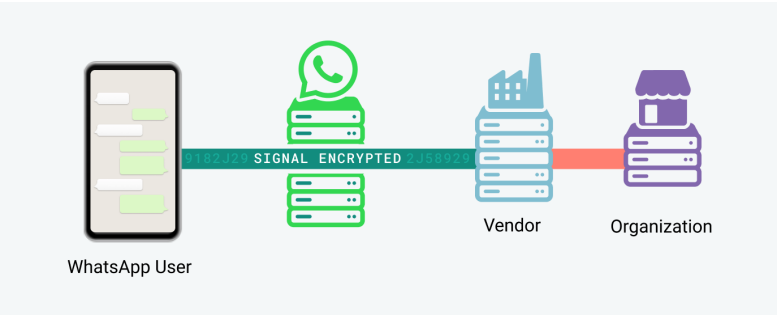


Algumas organizações podem usar a API do WhatsApp Business, um aplicativo que pode ser implantado como *endpoint* do WhatsApp em um servidor. A API do WhatsApp Business permite que essas organizações enviem e recebam mensagens de maneira programática.

O WhatsApp considera as comunicações com usuários da API do WhatsApp Business que gerenciam o *endpoint* da API em servidores que eles mesmos controlam como criptografadas de ponta a ponta, já que não há acesso de terceiros ao conteúdo entre os *endpoints*.



Algumas organizações podem optar por delegar a gestão do endpoint da API do WhatsApp Business a um prestador de serviço. Nesses casos, a comunicação ainda usa a mesma criptografia do Protocolo Signal e os clientes da versão v2.31 ou após esta são configurados para gerar chaves privadas no endpoint de API controlado pelo prestador de serviço. No entanto, como o usuário da API do WhatsApp Business escolheu um terceiro para gerenciar seu *endpoint*, o WhatsApp não considera essas mensagens como protegidas com a criptografia de ponta a ponta.



Em 2021, as organizações que usam a API do WhatsApp Business poderão designar o Facebook, empresa controladora do WhatsApp, como o prestador de serviço que opera o endpoint da API do WhatsApp Business em nome delas. Já que tais mensagens não são entregues diretamente a um *endpoint* controlado pela organização, o WhatsApp não considera conversas com organizações que optam pelo Facebook para operar o *endpoint* da API delas como protegidas com a criptografia de ponta a ponta.

Não é possível desativar a criptografia

Todas as conversas usam o mesmo Protocolo Signal, apresentado neste documento, que é independente do status da criptografia de ponta a ponta. O servidor WhatsApp não tem acesso às chaves privadas do cliente, entretanto, se um usuário do WhatsApp Business delegar a operação da sua API do WhatsApp Business a um prestador de serviço, esse prestador de serviço terá acesso às suas chaves privadas - mesmo que esse prestador de serviço seja o Facebook.

Ao conversar com uma empresa que usa a API do WhatsApp Business, o WhatsApp determina o status de criptografia de ponta a ponta com base apenas na escolha da empresa de quem opera seu *endpoint*.

O status de criptografia de uma conversa criptografada de ponta a ponta não pode ser alterado sem que a alteração seja visível para o usuário.

Exibição do status da criptografia de ponta a ponta

Em todos os nossos serviços, o WhatsApp deixa claro o status de criptografia de ponta a ponta de uma conversa. Se o telefone do usuário verificar que ele está se comunicando com um *endpoint* da API que delega a operação da API a um prestador de serviço, o telefone exibirá essa informação para o usuário. O usuário também pode verificar novamente o status diretamente na conversa ou na seção de informações comerciais do app.

Estas mudanças entrarão em vigor em todas as versões do WhatsApp após janeiro de 2021.

Conclusão

Todas as mensagens do WhatsApp são enviadas com o mesmo Protocolo Signal descrito acima. O WhatsApp considera todas as mensagens de um dispositivo controlado pelo remetente para aquele cujo endpoint é controlado pelo destinatário como criptografadas de ponta a ponta. As comunicações com um destinatário que opta por utilizar um fornecedor para gerenciar seu endpoint da API não são consideradas criptografadas de ponta a ponta. Nesse caso, o WhatsApp informará os usuários diretamente na conversa.

A biblioteca do Protocol Signal usada pelo WhatsApp é baseada na biblioteca Open Source, disponível aqui: http://github.com/whispersystems/libsignal-protocol-java/