WhatsApp

Visão Geral da Criptografia do WhatsApp

Documento Técnico

Versão 6 atualizada em 15 de novembro de 2021
Pré-visualização da versão 5 (aplicável apenas para versão beta para múltiplos aparelhos) atualizada em 27 de setembro de 2021
Pré-visualização da versão 4 (aplicável apenas para versão beta para múltiplos aparelhos) atualizada em 14 de julho de 2021
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

Segurança das mensagens

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 usam 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. Devido à natureza efêmera das chaves criptográficas, mesmo quando as chaves de criptografia atuais do aparelho de um usuário estão fisicamente comprometidas, elas não podem ser usadas para descriptografar mensagens transmitidas anteriormente.

Um usuário pode ter vários aparelhos, cada um com seu próprio conjunto de chaves de criptografia. Se as chaves de criptografia de um aparelho forem comprometidas, um hacker não poderá usá-las para descriptografar as mensagens enviadas a outros aparelhos, mesmo que estejam registrados para o mesmo usuário. O WhatsApp também usa criptografia de ponta a ponta para criptografar o histórico de mensagens transferido entre aparelhos quando um usuário registra um novo aparelho.

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

Termos

Tipos de aparelho

- Aparelho principal: um aparelho que é usado para registrar uma conta do WhatsApp com um número de telefone. Cada conta do WhatsApp está associada a um único aparelho principal. Este aparelho principal pode vincular aparelhos complementares adicionais à conta. As plataformas dos aparelhos principais são compatíveis com Android e iPhone.
- Aparelho complementar: um aparelho que está vinculado a uma conta do WhatsApp existente pelo aparelho principal da conta. As plataformas de aparelhos principais, como iPhone e Android, não podem estar vinculados como um aparelho complementar.

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.

Outros tipos de chave

- `Chave secreta de vinculação` - um valor de 32 bytes que é gerado em um aparelho complementar e deve ser passado por um canal seguro ao aparelho principal, usado para verificar um HMAC da carga vinculada recebida do aparelho principal. A transmissão dessa chave de aparelhos complementares para o aparelho principal é feita pela leitura de um código QR.

Vinculação complementar

- Metadados de vinculação - um blob codificado de metadados atribuído a um aparelho complementar durante a vinculação, usado em conjunto com a `Chave de Identidade` do aparelho complementar para identificar um complementar vinculado nos clientes do WhatsApp.
- Dados da Lista de Aparelhos Conectados - uma lista codificada identificando os aparelhos complementares vinculados atualmente no momento da assinatura. Assinado pela `Chave de identidade` do aparelho principal usando o prefixo 0x0602.
- Assinatura de conta - uma assinatura Curve25519 computada sobre o prefixo 0x0600, metadados de vinculação e a `Chave de Identidade` pública do aparelho complementar usando a `Chave de Identidade` do aparelho principal.
- Assinatura do aparelho - uma assinatura Curve25519 computada sobre o prefixo 0x0601, metadados de vinculação e a `Chave de Identidade` pública do aparelho complementar e a `Chave de Identidade` pública dos aparelhos principais usando a `Chave de Identidade` de um aparelho complementar.

Registro do cliente

Registro do aparelho principal

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.

Registro do aparelho complementar

Para vincular um aparelho complementar à conta do WhatsApp, o aparelho principal do usuário primeiro tem que criar uma assinatura de conta assinando a `Chave de Identidade` pública do novo aparelho e o aparelho complementar deve criar uma Assinatura de aparelho assinando a `Chave de Identidade` pública do aparelho principal. Uma vez que ambas as assinaturas forem produzidas, sessões protegidas com a criptografia de ponta a ponta podem ser estabelecidas com o aparelho complementar.

Para vincular um aparelho complementar:

1. O cliente complementar exibe sua chave de identidate pública $(I_{companion})$ e uma chave Secreta de Vinculação efêmera $(L_{companion})$ gerada em um código QR de vinculação. $L_{companion}$ nunca é enviado ao servidor do WhatsApp.
2. O cliente principal lê o código QR de vinculação e salva o $I_{companion}$ no disco.
3. O principal carrega sua própria Chave de Identidade como $I_{primary}$.
4. O principal gera Metadados de Vinculação como $L_{metadata}$ e Dados atualizados da lista de aparelhos contendo o novo complementar como $L_{istData}$.
5. O principal gera uma Assinatura de Conta para o complementar, $A_{signature} = CURVE25519\_SIGN(I_{primary}, 0x0600 || L_{metadata} || I_{companion})$.
6. O principal gera uma Assinatura da Lista do Aparelho para os Dados da Lista de Aparelhos atualizados, $ListSignature = CURVE25519\_SIGN(I_{primary}, 0x0602 || ListData)$.
7. O principal faz a série dos dados de vinculação ($L_{data)}$ contendo $L_{metadata}$, $I_{primary}$ e $A_{signature}$.
8. O principal gera uma vinculação $HMAC, PHMAC = HMAC\text{-}SHA256(L_{companion}, L_{data})$.
9. O principal envia $ListData, ListSignature, L_{data}$ e PHMAC para o servidor do WhatsApp. Consulte “Segurança do transporte” para informações sobre a conexão segura entre os clientes do WhatsApp e os servidores.

10. O servidor armazena `ListData` e `ListSignature`, e encaminha L_{data} e `PHMAC` para o complementar.

11. O complementar verifica `PHMAC`, decodifica `Ldata` em `Lmetadata`, `Iprimary` e `Asignature`, e verifica `Asignature`.

12. O complementar salva $L_{metadata}$ e $I_{primary}$ para o disco.

13. O complementar gera uma Assinatura de Aparelho pata ele mesmo, $D_{signature} = CURVE25519\_SIGN(I_{companion}, 0x0601 || L_{metadata} || I_{companion} || I_{primary})$.

14. O complementar carrega o $L_{metadata}$, $A_{signature}$, $D_{signature}$, $I_{companion}$, a Pré-chave assinada pública do complementar (com a assinatura), e um lote de Pré-chaves de Uso Único públicas para o servidor do WhatsApp.

15. O servidor armazena os dados carregados associados ao identificador do usuário combinado com um identificador específico do aparelho.

Configuração da Sessão Inicial

Para que os usuários do WhatsApp se comuniquem entre si em segurança e com privacidade, o cliente remetente estabelece uma sessão criptografada em pares com cada um dos aparelhos do destinatário. Além disso, o cliente remetente estabelece uma sessão criptografada em pares com todos os outros aparelhos associados à conta do remetente. Uma vez estabelecidas essas sessões protegidas com criptografia em pares, os clientes não precisam reconstruir novas sessões com esses aparelhos a menos que a sessão seja perdida, o que pode acontecer por causa de um evento como uma reinstalação do app ou uma mudança de aparelho.

O WhatsApp usa essa abordagem de "client-fanout" para transmitir mensagens para vários aparelhos, em que o cliente do WhatsApp transmite uma única mensagem N número de vezes para N número de diferentes aparelhos. Cada mensagem é criptografada individualmente usando a sessão de criptografia estabelecida em pares com cada 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 cada aparelho do destinatário e cada aparelho adicional do usuário que inicia a sessão (exceto o iniciador).

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. Além disso, para cada cada aparelho complementar (tanto para a conta do iniciador quanto para a do destinatário), o servidor também devolve os Metadados de

Vinculação ($L_{metadata}$), Assinatura da Conta ($A_{signature}$) e a Assinatura do Aparelho ($D_{signature}$) que foi carregada pelo aparelho complementar quando vinculado.

3. Para cada conjunto de chaves devolvidas para um aparelho complementar, o iniciador precisa verificar a `Assinatura A` por $CURVE25519\_VERIFY\_SIGNATURE(I_{primary}, 0x0600 || L_{metadata} || I_{companion})$, e $D_{signature}$ por $CURVE25519\_VERIFY\_SIGNATURE(I_{companion}, 0x0601 || L_{metadata} || I_{companion} || I_{primary})$. Se uma das verificações falhar para um aparelho complementar, o iniciador encerra imediatamente o processo de construção da sessão de criptografia e não envia nenhuma mensagem a esse aparelho.

Depois de obter as chaves do servidor e verificar a identidade de cada aparelho, o iniciador começa a estabelecer a sessão de criptografia com cada aparelho individual:

1. O iniciador salva a `Chave de Identidade` do destinatário como $I_{recipient}$, a `Pré-chave assinada como` $S_{recipient}$, e a `Pré-chave de Uso Único como` $O_{recipient}$.
2. O iniciador gera um par de chave Curve25519 efêmero, $E_{initiator}$.
3. O iniciador carrega a própria `Chave de Identidade` como $I_{initiator}$.
4. O iniciador calcula um segredo mestre como $master\_secret = ECDH(I_{initiator}, S_{recipient}) || ECDH(E_{initiator}, I_{recipient}) || ECDH(E_{initiator}, S_{recipient}) || ECDH(E_{initiator}, O_{recipient})$. Se não houver uma `Pré-chave de Uso Único`, o ECDH final será omitido.
5. 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 ($E_{initiator}$ e $I_{initiator}$) Além disso, se o iniciador for um aparelho complementar, ele também inclui $I_{primary}$, $L_{metadata}$, $A_{signature}$ e $D_{signature}$.

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

1. Se o remetente é um aparelho complementar, o destinatário precisa verificar a $A_{signature}$ por CURVE25519_VERIFY_SIGNATURE ($I_{primary}$, 0x0600 ||

$L_{metadata}$ || $I_{companion}$), e $D_{signature}$ por CURVE25519_VERIFY_SIGNATURE ($I_{companion}$, 0x0601 || $L_{metadata}$ || $I_{companion}$ || $I_{primary}$). Se uma das verificações falhar, o destinatário para de construir a sessão de criptografia e rejeita a mensagem daquele aparelho remetente.

2. O destinatário calcula o master_secret correspondente usando as próprias chaves privadas e as chaves públicas anunciadas no cabeçalho da mensagem recebida.
3. O destinatário exclui a Pré-chave de Uso Único utilizada pelo iniciador.
4. 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. O cliente usa client-fanout para todas as mensagens trocadas, o que significa que cada mensagem é criptografada para cada aparelho com a sessão correspondente em pares.

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” e da ida e volta “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(`$\text{Ephemeral}_{\text{sender}}$`, `$\text{Ephemeral}_{\text{recipient}}$`)`.
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.

Consistência de Aparelhos na Conversa

Em conversas com criptografia de ponta a ponta, para cada mensagem enviada a uma sessão de criptografia em pares, incluindo aquelas enviadas durante a configuração da sessão, o remetente inclui informações sobre a lista de aparelhos do remetente e do destinatário dentro da carga criptografada. Essa informação inclui:

1. O carimbo de data e hora mais recente da Lista de Aparelhos Conectados do remetente
2. Uma sinalização indicando se o remetente tem algum aparelho complementar vinculado no momento
3. O carimbo de data e hora mais recente da Lista de Aparelhos Conectados do destinatário
4. Uma sinalização indicando se o destinatário tem algum aparelho complementar conhecido vinculado

Ao executar o “client-fanout” para seus próprios aparelhos, os itens 3 e 4 acima continuam a se referir ao destinatário da mensagem original.

Transmissão de arquivos de mídia e outros anexos

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

1. Quando o aparelho do 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 uma loja 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 na loja de blob.
5. Todos os aparelhos receptores descriptografam a mensagem, recuperam o blob criptografado da loja de blob, confirmam o hash SHA256, verificam o MAC e leem o texto não criptografado.

Mensagens em grupo

A criptografia de ponta a ponta das mensagens enviadas a grupos do WhatsApp usa as sessões protegidas com criptografia em pares, conforme descrito anteriormente na seção "Configuração da Sessão Inicial", para distribuir o componente "Chave de Remetente" do Protocolo de Mensagens de Sinal.

Quando uma mensagem é enviada a um grupo pela primeira vez, uma "Chave de Remetente" é gerada e distribuída para cada aparelho do grupo, usando as sessões protegidas com criptografia em pares. O conteúdo da mensagem é criptografado usando a "Chave de Remetente", alcançando uma *fan-out* eficiente e segura para as mensagens que são enviadas a grupos.

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.

As informações de Consistência de Aparelhos na Conversa são incluídas ao distribuir uma "Chave de Remetente" e depois excluídas das mensagens subsequentes criptografadas com a Chave de Remetente.

Preenchimento do lado do remetente

Conforme descrito acima, no WhatsApp, como cada mensagem é criptografada para cada aparelho com a sessão correspondente em pares, o cliente remetente deve especificar todos os aparelhos de destino no momento do envio. Os aparelhos que não estiverem listados no momento do envio não receberão a mensagem criptografada. Cada cliente mantém uma lista de aparelhos complementares verificados para as contas do WhatsApp com as quais o usuário se comunica, bem como todos os outros aparelhos associados à própria conta, e usa esta lista para especificar os aparelhos de destino no momento do envio.

No entanto, ao enviar uma mensagem, é possível que um cliente perca aparelhos complementares válidos se a lista de aparelhos estiver desatualizada. O mecanismo “Preenchimento do lado do remetente” foi desenvolvido para que estes aparelhos perdidos possam se recuperar da falta permanente de toda a mensagem. Quando o WhatsApp recebe a mensagem criptografada do remetente, ela compara o hash de todos os aparelhos de destino listados pelo remetente, com o hash dos registros de aparelhos do lado do servidor dessas contas. Se houver uma divergência entre os dois valores de hash, o servidor notificará o remetente para atualizar a lista de aparelhos para ele mesmo e para todas as contas do destinatário. O cliente remetente verificará todos os novos aparelhos complementares, estabelecerá as sessões em pares com esses aparelhos usando o mesmo método descrito na seção "Configuração da Sessão Inicial", criptografará e reenviará a mensagem original para esses novos aparelhos.

Para garantir a confidencialidade da mensagem, este mecanismo de preenchimento só é permitido dentro de 5 minutos após o envio da mensagem inicial. Além disso, a mensagem de preenchimento nunca será enviada a nenhum aparelho complementar que tenha falhado na verificação do aparelho. No entanto, durante o processo de reenvio, se um destinatário fizer registro em um novo telefone, todos os seus aparelhos complementares serão excluídos da lista de reenvio. Desta forma, a mensagem de reenvio nunca será enviada a nenhum aparelho complementar de um destinatário com a chave de identidade alterada.

Sincronização do histórico de mensagens

Imediatamente após vincular um aparelho complementar, o aparelho principal copia as mensagens de conversas recentes com a criptografia de ponta a ponta.

O aparelho principal também incluirá uma cópia da chave de identidade pública do usuário armazenada ao copiar mensagens para conversas individuais. Este processo, chamado "Sincronização do histórico de mensagens", gera pacotes de mensagens protegidas com a criptografia de ponta a ponta e outros dados para a conversa usando o mesmo mecanismo de criptografia descrito na seção "Transmissão de arquivos de mídia e outros anexos". Os passos de 1 a 5 explicam as especificidades relativas à chave, IV, geração de mac, bem como a criptografia, transmissão e decodificação desses pacotes protegidos com a criptografia de ponta a ponta. Uma vez que um aparelho complementar tenha descriptografado, desempacotado e armazenado todas as mensagens de um determinado pacote, todos os dados associados são excluídos do aparelho complementar (incluindo o blob do pacote criptografado baixado, o ponteiro para o armazenamento do blob criptografado e todas as chaves).

Configuração da chamada

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

1. O cliente iniciador ("iniciador") estabelece sessões protegidas criptografadas com cada um dos aparelhos do destinatário (como descrito na Seção de Configuração da Sessão Inicial), caso estes ainda não tenham sido configurados.

2. O iniciador gera um conjunto de segredos mestre `SRTP` aleatórios de 32 bytes para cada um dos aparelhos do destinatário.

3. O iniciador envia uma mensagem de chamada recebida para cada um dos aparelhos do destinatário. O aparelho de cada destinatário recebe esta mensagem, que contém o segredo mestre `SRTP` protegido com a criptografia de ponta a ponta.

4. Se o respondente atender a chamada de um dos aparelhos, uma chamada criptografada `SRTP` é iniciada, protegida pelo segredo mestre SRTP gerado para aquele aparelho.

O segredo mestre `SRTP` é mantido na memória do aparelho do cliente e é usado somente durante a chamada. Os servidores do WhatsApp não têm acesso aos segredos mestres `SRTP`.

Chamada em grupo

As chamadas do WhatsApp são protegidas com a criptografia de ponta-a-ponta. Ao contrário das chamadas individuais, que são configuradas apenas uma vez, nas chamadas em grupo, as chaves são redefinidas sempre que algum participante entra ou sai da chamada.

A redefinição da chave em chamadas em grupo é realizada da seguinte forma:

1. Quando um participante entra ou sai da chamada, o servidor do WhatsApp seleciona arbitrariamente um dos participantes ativos como o distribuidor principal.

2. O distribuidor principal gera um segredo mestre `SRTP` aleatório de 32 bytes.

3. O distribuidor principal estabelece uma sessão criptografada com o aparelho ativo de cada participante conectado na chamada em grupo atual (como descrito na Seção de Configuração da Sessão Inicial), caso ainda não tenham sido configuradas.

4. O distribuidor principal inicia uma mensagem com o segredo mestre `SRTP` protegido com a criptografia de ponta a ponta para cada participante. Quando as mensagens são entregues, acontece uma chamada em grupo `SRTP` criptografada.

Em uma chamada em grupo, um participante se torna ativo quando inicia uma chamada em grupo ou aceita o convite da chamada em grupo de qualquer um de seus aparelhos. Portanto, cada participante ativo tem exatamente um aparelho ativo.

O segredo mestre `SRTP` é mantido na memória e é substituído quando um novo segredo mestre `SRTP` é gerado e entregue. Os servidores do WhatsApp não têm acesso a eles e não podem acessar as mídias de áudio e vídeo reais.

Para garantir a qualidade da chamada e evitar condições conflitantes de corrida de usuários, o servidor do WhatsApp armazena o estado da chamada em grupo atual (por exemplo: lista de participantes, iniciador da chamada) e metadados de mídia (por exemplo: resolução, tipo de mídia). Com essa informação, o servidor do WhatsApp pode transmitir as mudanças dos participantes e selecionar um como distribuidor principal para iniciar a redefinição da chave.

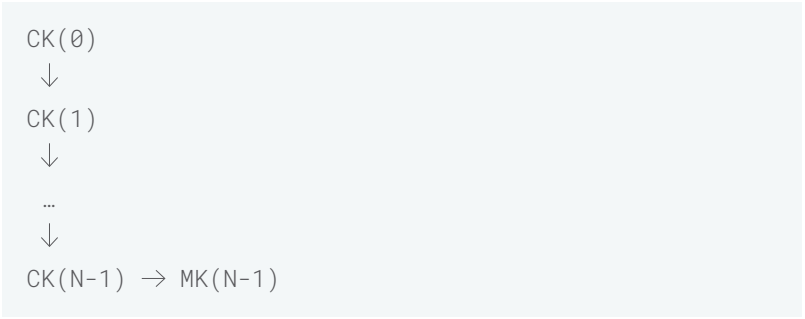
Status

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 aparelhos 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 aparelhos 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.

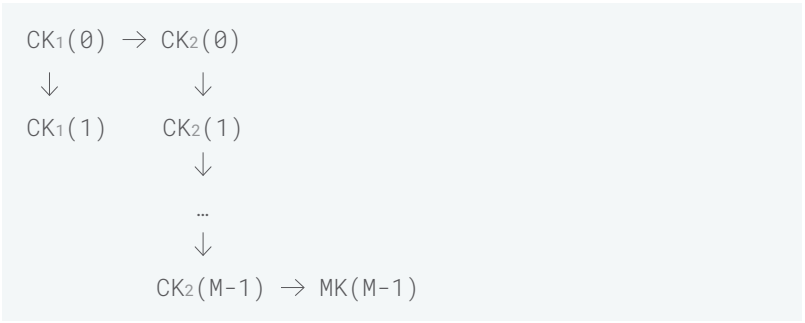
Localização em tempo real

As mensagens de localização em tempo real e as atualizações são criptografadas da mesma forma que as mensagens em grupo. No momento, enviar e receber a localização em tempo real só é compatível com aparelhos principais. 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, a localização em tempo real exige 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 o que for muito lento para esta aplicação. Este documento oferece um algoritmo rápido de ajuste para resolver este problema.

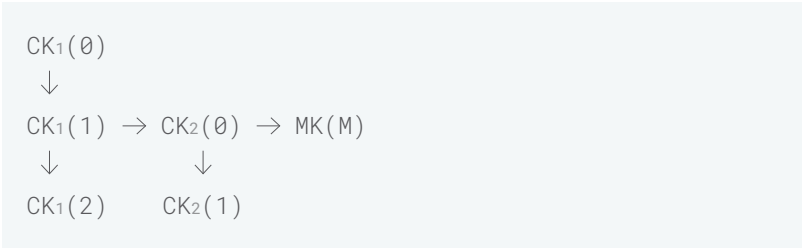
No momento, as Chaves Corrente são unidimensionais. Para ajustar `N` etapas, é preciso `N` cálculos. As Chaves Corrente são indicadas como `CK (contagem de iteração)` e Chaves de Mensagem como `MK (contagem de iteração)`.



Considere uma extensão onde guardamos duas séries de chaves corrente:

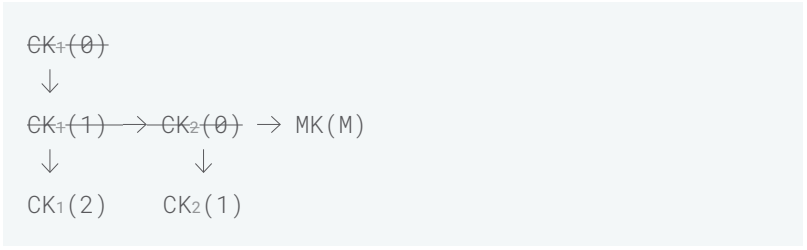


Neste exemplo, as Chaves de Mensagem são sempre derivadas de `CK₂`. Um destinatário que precisa de muitos ajustes pode pular `M` iterações de cada vez (em que `M` é um inteiro positivo constante estabelecido) pelo ajuste do `CK₁` e gera um novo `CK₂`:



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

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.

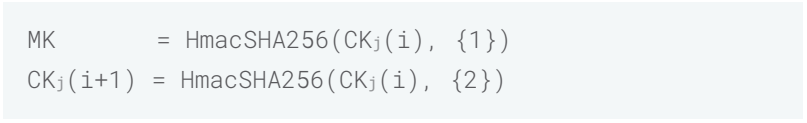
Os valores legais para D são potências positivas de dois menores ou iguais ao número de bits na contagem de iteração: 1, 2, 4, 8, 16, e 32. Os implementadores selecionam um valor de D como uma troca de memória explícita da CPU (ou largura de banda da rede da CPU).

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 leva mais que $D \times M$ etapas.

O Signal utiliza funções diferentes para o cálculo do ajuste 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:

$$j = 1 : CK_1(0) \coloneqq RNG(32)$$
$$j > 1 : CK_j(0) \coloneqq HmacSHA256(CK_{j-1}(0), \{j+1\})$$

E ajustadas como:

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

Segurança da Sincronização do Estado do App

Apresentação

A Sincronização do Estado do App permite uma experiência consistente entre aparelhos, e é protegida com a criptografia de ponta a ponta. Antes de ser compatível com aparelhos complementares, o cliente do WhatsApp era o único proprietário e a fonte confiável para todas as configurações do cliente e outros dados, referido como Estado do App. Com a implantação dos aparelhos complementares, o Estado do App é sincronizado com todos os aparelhos do usuário com segurança, usando a criptografia de ponta a ponta.

Exemplos das configurações do cliente da Sincronização do Estado do App e outros dados incluem o seguinte:

- Propriedades da conversa, como Silenciar, Fixar, Apagar
- Propriedades da mensagem, como Apagar para mim, Favoritar
- Propriedades relacionadas a contatos, tais como nomes de contatos, nomes de listas de transmissão
- GIFs, figurinhas e emojis usados mais recentemente

O Estado do App não inclui o conteúdo das mensagens do usuário, nem chaves que poderiam ser usadas para descriptografar mensagens, nem configurações que possam impactar o sigilo das mensagens.

A sincronização do Estado do App entre os aparelhos de um usuário requer o armazenamento de dados protegidos com a criptografia de ponta a ponta no servidor do WhatsApp para facilitar a transmissão entre os diferentes aparelhos para a conta do usuário. Os servidores do WhatsApp não têm acesso às chaves que poderiam ser usadas para descriptografar os dados que são armazenados no Estado do App.

Uma **Coleção** é uma representação de vários casos de uso agrupados. Por exemplo, várias Propriedades da conversa (ver acima) podem ser modeladas como uma única Coleção. As Coleções são implementadas como um dicionário (um conjunto de pares de valores indexados) e são fixadas para versões específicas do cliente.

De início, as Coleções estão vazias. Para modificar uma Coleção, um aparelho cliente apresenta uma **Mutação** que ou define um novo Valor para um determinado Índice (DEFINIR Mutação), ou remove o par da Coleção (REMOVER Mutação).

Um grupo de Mutações enviadas juntas constitui um **Patch**, que muda atomicamente a Coleção da versão N para a versão N + 1. O servidor mantém uma fila ordenada de Patches (**Fila de Patches**), que é composto por todos os Patches enviados nos últimos X dias.

Um processo do lado do servidor, chamado **Base Roller**, constrói periodicamente um **Snapshot** que representa o estado da Coleção depois de aplicar todos os Patches, incluindo o Patch com a versão N. O primeiro Snapshot é construído a partir de toda a Fila de Patches, enquanto os Snapshots subsequentes são construídos aplicando novos Patches ao Snapshot anterior. Um Snapshot pode ser usado para inicializar um aparelho recentemente registrado ou para otimizar o tráfego de dados enviando um Snapshot ao invés da lista de Patches.

A Sincronização do Estado do App foi desenvolvida para garantir o sigilo e a integridade dos dados que estão sendo sincronizados. As sessões protegidas com criptografia em pares (conforme descrito na Seção de Configuração da Sessão Inicial) são usadas para transferir chaves secretas entre diferentes aparelhos da mesma conta.

Termos

- `Chave de Base` - material da chave de entrada usado para gerar as chaves usadas para criptografar os dados ou fornecer a integridade dela.
- `Chave de Índice MAC` - chave derivada da `Chave de Base` via HKDF e usada para calcular o HMAC do índice.
- `Chave de Criptografia de Valor` - chave derivada da `Chave de Base` via HKDF e usada para criptografar o índice e o valor combinados da Mutação. A criptografia é feita via AES-256 no modo CBC.
- `Chave MAC de Valor` - chave derivada da `Chave de Base` via HKDF e usada para calcular o `HMAC` do índice e valor combinado de Mutação. Usado no estágio MAC da abordagem `Encrypt-then-MAC` para fornecer criptografia autenticada.
- `Chave MAC de Snapshot` - chave derivada da `Chave de Base` via HKDF e usada para fornecer antiviolação para Snapshots gerados pelo Base Roller.
- `Chave MAC de Patch` - chave derivada da `Chave de Base` via HKDF e usada para fornecer antiviolação para Patches.
- `KeyID` - Identificador exclusivo para a `Chave de Base`. As `Chaves de Base` são giradas periodicamente e quando um aparelho é removido da conta para fornecer um eventual futuro sigilo. Um hacker em posse de um aparelho removido e acesso ao servidor não pode mais descriptografar o conteúdo das `Mutações SET` enviado após a remoção.
- `Operação` - Valor de byte que identifica uma Mutação como `DEFINIR` ou `REMOVER`.

Criptografia das Mutações

Para que o Base Roller possa unir as sequências de ações ao mesmo índice, ele precisa que o índice enviado ao servidor seja determinístico. O HMAC do índice é usado como um identificador do registro do valor do índice a que as Mutações se referem. Isto também garante que os índices que o servidor vê tenham o mesmo comprimento e evita que o servidor adivinhe o qual o registro em que a Mutação foi usada.

Os valores (junto com os índices, conforme mencionado acima) são criptografados usando a criptografia padrão autenticada (descrita abaixo) com IVs aleatórios.

O índice combinado e o texto não criptografado de valor são complementados com almofadas de comprimento arbitrário para permitir um modelo de privacidade diferencial sobre o tipo de registro.

1. Gere a `Chave MAC do Índice, Chave de Criptografia de Valor, Chave MAC de Valor, Chave MAC do Snapshot` e a `Chave MAC de Pach` da `Chave de Base` por meio de HKDF.
2. Calcule o HMAC-SHA2-256 do índice.
3. Construa o texto não criptografado, combinando o Índice e o Valor com preenchimento aleatório (usado para ofuscar o tamanho da Mutação do servidor).
4. Construa os dados associados, concatenando a `Operação` com a `KeyID`.
5. Aplique a abordagem `Encrypt-then-MAC` com `AES-256-CBC` chaveada por `Chave de criptografia de Valor` e `HMAC-SHA2-512` chaveada por `Chave MAC de Valor`.
6. MAC computado no Passo 2, texto codificado computado no Passo 5, junto com a `Operação` e a `KeyID` formam uma Mutação criptografada.

Antiviolação

Os mecanismos de antiviolação descritos abaixo foram criados para evitar:

- Soltar, reordenar ou repetir a Mutação dentro de um Patch
- Soltar, reordenar ou reproduzir (incluindo a construção de novos Patches) Patches dentro de uma Coleção ou mesmo entre a Coleção
- Soltar ou repetir as Mutações dentro de um Snapshot construído por um Base Roller

Integridade do Snapshot

O servidor executa periodicamente um Base Roller que compacta a Fila de Patch em um único Snapshot. Os clientes não podem prever quando o Snapshot será construído (no extremo, os Snapshots poderiam ser construídos em todos os Patches). Assim, os clientes incluem uma soma de verificação adicional e impossível de falsificar para cada Patch, a fim de conseguir verificar todos os Snapshots possíveis construídos pelo servidor.

Nossa abordagem depende de um algoritmo de hashing homomórfico chamado LtHash. Ele tem as duas seguintes importantes propriedades:

- Homomorfismo de grupos: para para dois grupos separados `S` e `T`, `LtHash(S) + LtHash(T) = LtHash(S U T)`.
- Resistência à colisão: é difícil (computacionalmente inviável) encontrar dois conjuntos distintos `S` e `T` para os quais `LtHash(S) = LtHash(T)`.

Uma variante de 1024 bits chamada `LtHash16` e HKDF é usada como uma função de saída extensível (XOF). Para cada Coleção os clientes devem manter um valor de 1024 bits do `LtHash16` calculado sobre o Snapshot atual da Coleção. O MAC computado sobre o conteúdo do índice de texto não criptografado e o valor junto com os dados autenticados é usado como entrada para o `LtHash`.

Ao receber ou construir um novo Patch, a propriedade de homomorfismo de grupos do `LtHash` é usado para calcular o novo valor de 1024 bits correspondente ao novo estado (após a aplicação do Patch em questão):

- Supondo que o valor atual da compilação seja `CurrentLtHash`, e o Patch P esteja sendo processado.
- Construa um grupo `R` de `MACs` de estados anteriores de todos os registros afetados (excluídos com REMOVER ou substituídos com uma operação DEFINIR).
- Construa um grupo `A` de `MACs` de todos os registros DEFINIR (em forma criptografada) no Patch `P`.
- Construa `NewLtHash = LtHash16Add(LtHash16Subtract (CurrentLtHash, R), A)`. Veja abaixo como essas operações são definidas.

`A operação LtHash16Add` mencionada acima é definida da seguinte maneira:

```
LtHash16Add(H, A):
    R = H
    para o Item em A:
        R = LtHash16AddSingle(R, Item)
    retorna R

LtHash16AddSingle(H, I):
    X = HKDF(1024, “Integridade do Patch do
WhatsApp”, I)
    retorna PointwiseAdd16(H, X)
```

onde `PointwiseAdd16` realiza pontualmente a adição de duas matrizes de bytes de 1024 bits, interpretando-as como matrizes de inteiros de 16 bits sem assinatura. A operação `LtHash16Subtract` é definida de maneira similar à `LtHash16Add` substituindo a adição pontual por subtração pontual.

Além disso, um MAC sobre o valor computado do `LtHash16` concatenado com a versão do Patch de 8 bytes e o nome da Coleção é computado:

```
SnapshotMAC = HMAC_SHA_256(
    SnapshotMACKey,
    LtHash ||
    TO_64_BIT_NETWORK_ORDER(PatchVersion) ||
    TO_UTF8(CollectionName)
)
```

Integridade da Fila de Patch

Para evitar a violação com o conteúdo de um Patch, os clientes devem calcular o HMAC sobre os 32 bytes de MAC de cada Mutação individual que faz parte do Patch, juntamente com o número da versão do Patch e do Nome da Coleção:

```
PatchMAC = HMAC_SHA_256(
    PatchMACKey,
    SnapshotMAC ||
    MutationMAC_0 ||
    MutationMAC_1 ||
    .. ||
    MutationMAC_N ||
    TO_64_BIT_NETWORK_ORDER(PatchVersion) ||
    TO_UTF8(CollectionName)
)
```

onde `MutationMAC_i` são os últimos 32 bytes do valor de texto codificado da Mutação #i no Patch, e `PatchVersion` é a versão do Patch prestes a ser enviada, ou seja, a última versão conhecida da Coleção mais uma. Note que, ao receber um Patch, o cliente deve verificá-lo, incluindo a versão esperada do Patch, que deve corresponder à versão atribuída ao servidor.

Tanto o valor `PatchMAC` quanto o `SnapshotMAC` estão incluídos no Patch e são enviados ao servidor.

Verificação

Depois de baixar um Patch, os clientes devem primeiro verificar sua correção, recalculando o `PatchMAC` e comparando-o com o valor incluído com o Patch. Depois disso, os clientes verificam se o `SnapshotMAC` também está correto, repetindo os passos descritos acima.

O processo de Base Roller no servidor deve preservar o `SnapshotMAC` (e o `KeyID` usado para gerá-lo) do último Patch que foi usado para construir o Snapshot do Base Roller. Este valor é usado por um cliente que recebeu um Snapshot para verificar, de maneira independe, a integridade, aplicando o LtHash16 sobre todos os registros dele e calculando o MAC conforme descrito acima.

Rotação de Chave

Uma Rotação de Chave envolve um cliente gerando aleatoriamente um novo tuple de chave e transmitindo-o a todos os outros aparelhos. No caso da Rotação de Chave, todas as futuras Mutações não devem usar nenhuma versão de chave anterior. Para preservar a capacidade do servidor de unir as Mutações aplicadas a um registro durante a atualização em todo limite da chave, um cliente deve enviar uma Mutação `REMOVER` com a chave antiga e uma Mutação `DEFINIR` (se necessário) com a nova chave. Notamos que a ocorrência simultânea de `REMOVER` e `DEFINIR` à medida que a versão chave aumenta, fica relativamente fácil para o servidor se correlacionar como equivalente à uma atualização do registro. Junto com qualquer pessoa com

acesso à chave antiga, o WhatsApp seria, portanto, capaz de determinar com alta confiança, o valor do novo índice e seria capaz de assumir que isso significa que ele foi atualizado.

Os dois mecanismos a seguir são usados para combater:

1. Ofuscação de Atualização Pós-Rotação - ao enviar uma Mutação ao servidor, os clientes usarão esta oportunidade para rotacionar outros números de registros, garantindo que o WhatsApp não consiga determinar quais índices antigos estavam sendo atualizados, e não possa mapear diretamente nenhum dos registros antigos para o qual o novo registro está representando.
2. Captura Assíncrona de Chaves - garante que, após uma Rotação de Chave, haverá um momento no futuro em que nenhum registro atual será criptografado com as chaves anteriores. Isso significa que, em alguma cadência, os clientes emitirão uma série de `DEFINIR` e `REMOVER` para criptografar novamente os registros antigos sob uma nova versão de chave, sem atualizar os valores reais de textos não criptografados. As atualizações de Catch-Up são indistinguíveis de uma operação lógica de `ATUALIZAÇÃO`, de modo que o servidor junto com um aparelho removido nunca pode determinar quando um registro antigo está sendo atualizado.

A chave deve ser rotacionada sempre que um aparelho estiver deixando de ser registrado. Além disso, os clientes rotacionam a chave periodicamente (uma vez por mês, por exemplo).

Cada dispositivo mantém uma lista das chaves de criptografia juntamente com dados adicionais:

1. `KeyData` - bytes da chave de base reais
2. `KeyID` - ID da chave
3. `Impressão digital` - Estrutura de dados que identifica uma lista de dispositivos existentes no momento em que a chave foi gerada e, desta forma, compartilhada
4. `Carimbo de data/hora` - hora em que a chave foi criada

O `KeyID` é composto e consiste de um `Epoch` de 4 bytes e um `DeviceID` de 2 bytes. O `Epoch` é selecionado aleatoriamente entre 1 e 65536 pelos aparelhos principais durante o registro do primeiro aparelho complementar. Depois disso aumenta em 1 cada vez que um aparelho rotaciona uma chave. O componente `DeviceID` do `KeyID` é usado para resolver corridas entre vários aparelhos que rotacionam a chave ao mesmo tempo, de modo que todas as chaves recebam `IDs` únicos. Para estabelecer em uma única chave após tal evento, os clientes preferem a chave com o menor componente `DeviceID` quando os componentes `Epoch` são iguais. Se não, prefira sempre o `KeyID` com o maior `Epoch`. Além disso, uma ou, em casos raros, várias chaves de criptografia podem estar ativas a qualquer momento.

A Rotação de Chave deve acontecer sob as seguintes condições:

- Se um cliente detectar que um aparelho previamente conhecido foi removido, todas as chaves de criptografia ativas devem ser marcadas localmente como expiradas.
- Ao receber uma mensagem `AppStateSyncKeyShare`, marque todas as chaves com o menor Epoch como expiradas.

- Ao receber uma Mutação em qualquer Coleção, marque todas as chaves com o menor Epoch como expiradas.
- Quando um cliente quiser enviar um novo Patch para o servidor, primeiro é preciso verificar a lista de chaves conhecidas. Ele usará aquela que ainda estiver ativa, se houver. Caso contrário, é preciso realizar a Rotação de Chave. Para rotacionar a chave:
    1. Gere um novo `KeyID` ao concatenar o `DeviceID` com um `Epoch` incremental (valor máximo entre todas as chaves de criptografia conhecidas).
    2. Gere um novo material chave a partir de `CSPRNG`.
    3. Gere uma nova `Impressão Digital` a partir dos dados de registro atuais.
    4. Mantenha e envie as informações-chave a todos os outros aparelhos usando as sessões correspondentes criptografadas em pares.
    5. Use a chave para criptografar e enviar as Mutações para o servidor.
- Em alguns casos, os clientes não conseguem determinar se uma chave ainda é válida com base apenas no pedido do evento. Para calcular se a última chave ativa conhecida é válida ou não, os clientes comparam a `Impressão Digital` da chave com os dados atuais de registro do aparelho.

Se um aparelho receber uma Mutação do servidor e o `KeyID` não for conhecido, um aparelho pode solicitar o reenvio das chaves de criptografia de outros aparelhos.

Para garantir que as chaves de criptografia não sejam compartilhadas com aparelhos não confiáveis, todas as aplicações de clientes só são enviadas por sessões criptografadas em pares autenticados:

1. Ao realizar a Rotação da Chave, um aparelho deve enviar a nova chave para todos os outros aparelhos, conhecidos por serem autorizados pelo principal.
2. Quando um aparelho recebe uma nova chave de outro aparelho que não é autorizado pelo principal, esta chave é ignorada.

Para garantir que outros aparelhos não usem inadvertidamente uma chave de criptografia que deve ser expirada na remoção do aparelho, o aparelho que realiza a remoção (o próprio aparelho complementar ou o principal) apresenta um Patch em todas as Coleções marcando todas as chaves atuais como expiradas. Este Patch informa outros aparelhos que as chaves de criptografia com epoch menor ou igual ao epoch fornecido não devem ser usadas no futuro.

Verificação das chaves

Os usuários do WhatsApp também têm a opção de verificar as chaves de seus aparelhos e de aparelhos de outros usuários que estão se comunicando por conversas protegidas com a 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). A verificação pode

ser feita pela leitura do código QR ou comparando o número de 60 dígitos entre dois aparelhos principais. Os usuários do WhatsApp também podem verificar manualmente os aparelho complementares individuais usando um aparelho principal para verificar o mesmo código QR ou 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 todos os aparelhos de ambas as partes.

Quando um aparelho 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 aparelho do usuário. Para calcular uma impressão digital numérica de 30 dígitos:

1. Classifique e concatene lexicograficamente as chaves de identidade pública para todos os aparelhos do usuário.
2. O hash SHA-512 combina iterativamente a `Chave de Identidade` e o identificador de usuário 5.200 vezes.
3. Pegue os primeiros 30 bytes da combinação final de saída.
4. Divida o resultado de 30 bytes em seis partes de 5 bytes.
5. 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.
6. Concatene os seis grupos de cinco dígitos em trinta dígitos.

Remoção do aparelho complementar

Os aparelhos complementares podem se desconectar de uma conta do WhatsApp, ser desconectados pelo aparelho principal do usuário ou ser desconectados pelo servidor do WhatsApp. Quando um aparelho principal se desconecta ou detecta a desconexão de um ou mais aparelhos complementares, enquanto um ou mais complementares permanece vinculado, ele gera e carrega novos dados da Lista de Aparelhos Conectados, removendo o aparelho previamente autorizado.

Para atualizar a Lista de Aparelhos Conectados:

7. O primário detecta a remoção de um aparelho e carrega sua própria Chave de Identidade como $I_{primary}$.
8. O principal gera dados atualizados da Lista de Aparelhos contendo os aparelhos atualmente vinculados, como `ListData`.

9. O principal gera uma Assinatura da Lista do Aparelho para os Dados da Lista de Aparelhos atualizados, `ListSignature = CURVE25519_SIGN(`$I_{primary}$`, 0x0602 || ListData)`.
10. O principal envia `ListData` e `ListSignature`, para o servidor do WhatsApp. Consulte “Segurança do transporte” para informações sobre a conexão segura entre os clientes do WhatsApp e os servidores.

Mesmo que nenhuma remoção de aparelho tenha sido detectada, enquanto um ou mais complementares permanecem vinculados, os aparelhos principais carregarão periodicamente dados atualizados da Lista de Aparelhos Conectados seguindo os passos acima para produzir uma assinatura com um carimbo de data/hora atualizado.

Validade da Lista de Aparelhos Conectados

Em conversas protegidas com a criptografia de ponta a ponta, as Listas de Aparelhos Conectados expiram com um Tempo de Vida de 35 dias ou menos. Os clientes só enviarão e receberão mensagens e chamadas com o aparelho principal de uma conta com uma Lista de Aparelhos Conectados expirados . Ao receber uma Lista de Aparelhos Conectados atualizada com um carimbo de data/hora mais recente, os remetentes se comunicarão mais uma vez com os aparelhos complementares vinculados do usuário.

Ao receber os Dados de Consistência de Aparelhos na Conversa com um carimbo de data/hora atualizado para a lista de aparelhos do remetente, os aparelhos receptores reduzem a TTL da lista atual de aparelhos do remetente para 48 horas ou menos a partir do recebimento da mensagem. Para manter a confiabilidade da mensagem, a TTL reduzida não será aplicada até que o cliente receptor chegue on-line após a janela de 48 horas.

Compromisso do aparelho complementar

O Tempo de Vida das Assinaturas da Lista de Aparelhos e a Consistência de Aparelhos na Conversa revogam as listas de aparelhos associados conectados após 35 dias e 48 horas respectivamente. Se as chaves privadas de um aparelho complementar ficarem comprometidas, o aparelho comprometido não deve mais ser usado, e deve removido da conta. Os aparelhos são revogados automaticamente após a remoção para limitar o potencial de um terceiro que comprometeu o aparelho conivente com o WhatsApp para continuar a enviar e receber mensagens da conta previamente vinculada. O par de chaves de identidade do aparelho principal pode ser gerado novamente ao apagar e reinstalar o WhatsApp em seu aparelho principal para revogar todos os aparelhos imediatamente.

Segurança do transporte

A comunicação entre os clientes do WhatsApp e os servidores de conversa do WhatsApp é colocada em um canal criptografado separado usando Tubos de Ruído com Curve25519, AES-GCM e SHA256 da Estrutura de Protocolo de Ruídos 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.

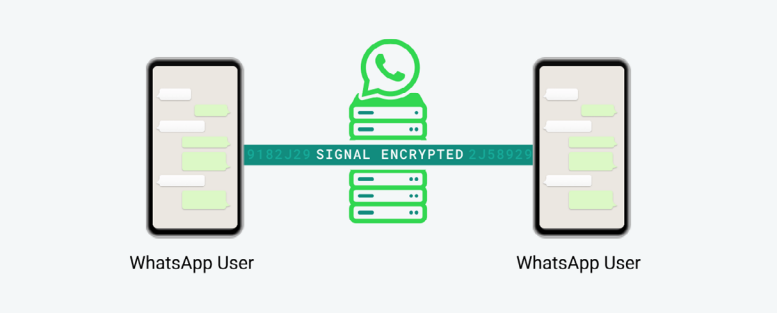
Nota: nos casos em que um usuário comercial delega a operação da API do WhatsApp Business do cliente a um fornecedor, esse fornecedor terá acesso às suas chaves privadas, inclusive se esse fornecedor for o Facebook. No entanto, essas chaves privadas ainda não serão armazenadas no servidor de conversas do WhatsApp. Para mais detalhes, consulte abaixo.

Definição da criptografia de ponta a ponta

O WhatsApp define criptografia de ponta a ponta como comunicações que permanecem criptografadas em um aparelho controlado pelo remetente para um aparelho 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

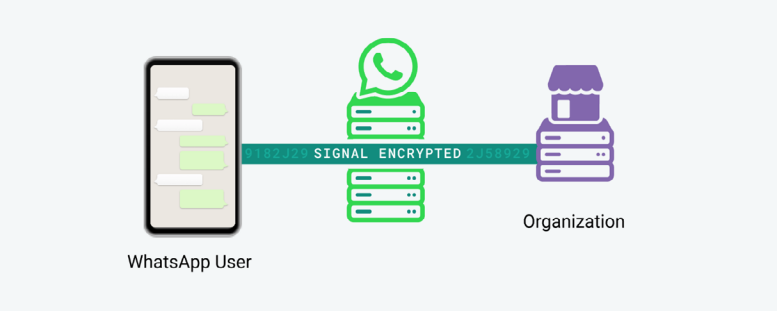
É muito simples quando se trata de duas pessoas que se comunicam em seus respectivos telefones ou computadores usando o app WhatsApp Messenger ou o WhatsApp Business: o endpoint do WhatsApp de cada pessoa funciona em um aparelho controlado pelas pessoas.



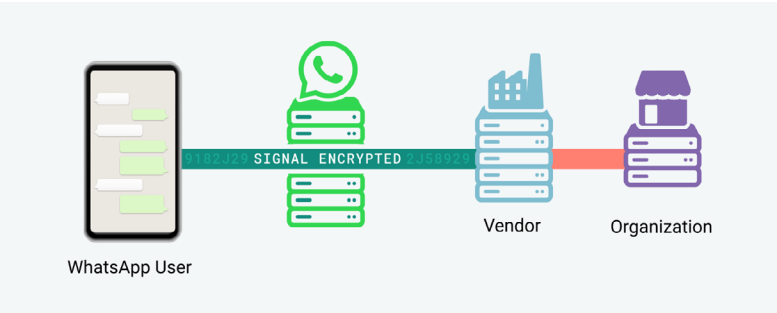
Algumas organizações podem usar a API do WhatsApp Business, um aplicativo que pode ser implantado como endpoint do WhatsApp em um servidor. A API

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 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 fornecedor. Nesses casos, a comunicação ainda usa a mesma criptografia de protocolo de sinal e os clientes na versão v2.31 ou mais recente são configurados para gerar chaves privadas dentro da API endpoint controlada pelo fornecedor. 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 fornecedor 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 do WhatsApp não tem acesso às chaves privadas do cliente, porém quando um usuário comercial delega a operação da API do WhatsApp Business do cliente a um fornecedor, esse fornecedor terá acesso às suas chaves privadas, inclusive se esse fornecedor for o Facebook.

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

O status de criptografia de uma conversa protegida com a criptografia de ponta a ponta não pode mudar sem que a mudança 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 fornecedor, o telefone exibirá essa informação para o usuário. O usuário também pode verificar novamente o status da criptografia diretamente na conversa ou na seção de informações comerciais do app.

Estas mudanças entrarão em vigor em todos as versões do WhatsApp depois de janeiro de 2021.

Conclusão

Todas as mensagens do WhatsApp são enviadas com o mesmo protocolo de sinal destacado acima. O WhatsApp considera todas as mensagens, chamadas de voz e chamadas de vídeo enviadas entre todos os aparelhos controlados por um usuário remetente e todos os aparelhos controlados por um usuário destinatário como protegidos com a criptografia de ponta a ponta. A sincronização do histórico de mensagens do WhatsApp e a sincronização do estado do app também são protegidas pela criptografia 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 Protocolo de Sinal usada pelo WhatsApp tem como base a biblioteca de Código fonte Aberto, disponível aqui:

http://github.com/whispersystems/libsignal-protocol-java/