😈

Criptografia: Parte 4

Técnica híbrida

Envia-se a mensagem cifrada com a chave simétrica, mas tambem envia-se a chave simétrica cifrada com a chave pública do destinatário.

Cenário email seguro

- Alice conhece a chave pública de Bob: o que permite a Alice enviar informação cifrada ao Bob.
- Bob conhece a chave de verificação da Alice (pública).

A Alice nunca vai ter certeza se Bob recebeu a mensagem ou não.

No caso simétrico nós primeiro ciframos e depois autenticamos, mas no caso do email seguro temos que primeiro **assinamos e depois autenticamos.**

Why not sign & encrypt

Se a Alice primeiro cifrar e depois assinar, a Alice pode afirmar que **não sabia qual era o conteúdo que estava a assinar.** O que anula a propriedade de **não repúdio.** Ainda, era possível que Alice envia uma mensagem para uma third-party (Carol). A carol poderia ler o conteúdo da mensagem e depois recifrá-la para o Bob. **Perdemos a confidencialidade.**

> Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML
>
> Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML Trust, but verify. - Russian proverb Simple Sign & Encrypt, by itself, is not very secure. Cryptographers know this well, but application programmers and standards authors still tend to put too much trust in simple Sign-and-Encrypt.
>
> @ https://theworld.com/~dtd/sign_encrypt/sign_encrypt7.html

Sessão 4.1 explica a perda de confidencialidade.

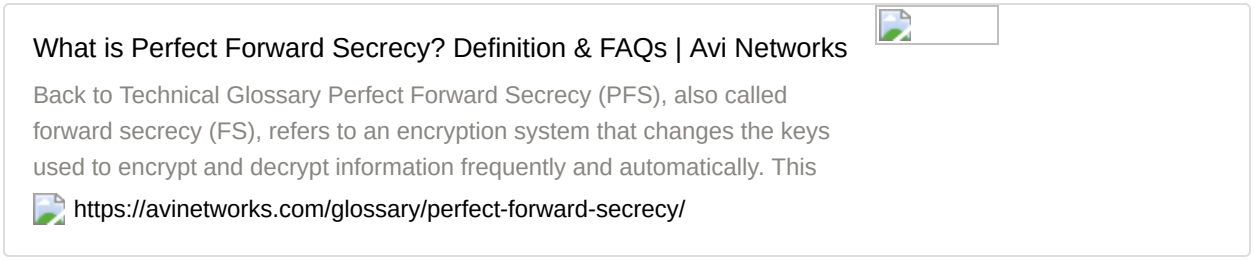
Metadados: precisamos ter cuidado com estes, porque definem também quem é o destinatário.

Cenário Acordo de Chaves

Pressupostos

Alice e Bob conhecem mutuamente as chaves públicas e de verificação de um do outro. (Ela já foi usada e se tem a confirmação disto).

Perfect forward secrecy



“Refers to an encryption system that changes the keys used to encrypt and decrypt information frequently and automatically. This ongoing process ensures that even if the most recent key is hacked, a minimal amount of sensitive data is exposed.”

Comprometer uma chave de longa duração não vai comprometer as sessões anteriores.

O que é um pouco difícil de atingir com uma cifra assimétrica. Porque se for revelada a chave que decifra a cifra e o atacante tiver o registo de todas as comunicações, ele consegue ter acesso aos registos anteriores. Por isso não usamos esta abordagem.

> *Curiosidade: demorou muitos anos pra se convergir para uma solução deste problema. Porque estas convergências são decididas pelo acordo de uma série de entidades. No entanto, estas entidades possuem seus próprios interesses como manter o código legacy que possuem dentro dos padrões de convenção. Por isso estes tipos de avanço demoram muito.*

Diffie-Hellman

> *Esta é uma intuição do protocolo.*

Parâmetros públicos: (G, g, o)

Conjunto G:

- Valores inteiros entre [0, p) é que p é um primo grande.
- Podem ser pontos numa curva elíptica: são pares de números inteiros entre [0,p).

Definimos um **p** e calculamos os pontos que satisfazem uma curva elíptica.

Operação o

Uma operação que permite combinar dois pontos numa curva e obter outro ponto na curva. São operações muito eficientes.

Gerador g

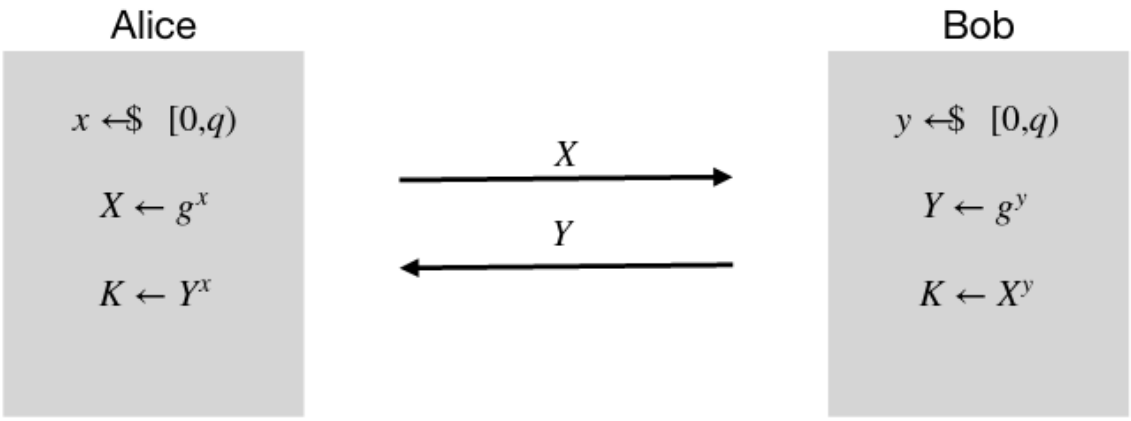
Por exponeciação conseguimos gerar muitos números.

Mas precisamos que a exponenciação tenha uma propriedade obvia:

$(g^x)^y = g^{xy} = g^{yx} = (g^y)^x$

Escolhemos um número primo **q** para elevar o **g.**

O protocolo

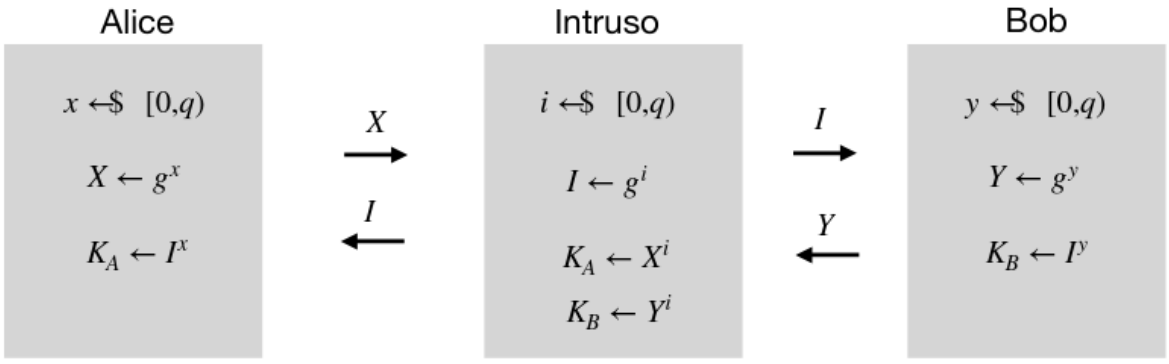


$$K = (g^y)^x = g^{yx} = g^{xy} = (g^x)^y = K$$

- O x e y são parâmetros privados. Mas X e Y são públicos e são trocados na rede.
- Com a troca de mensagens, ambos Bob e Alice aplicam a exponenciação da sua chave privada.

O ponto é que o atacante não pode conseguir fatorizar o X e o Y.

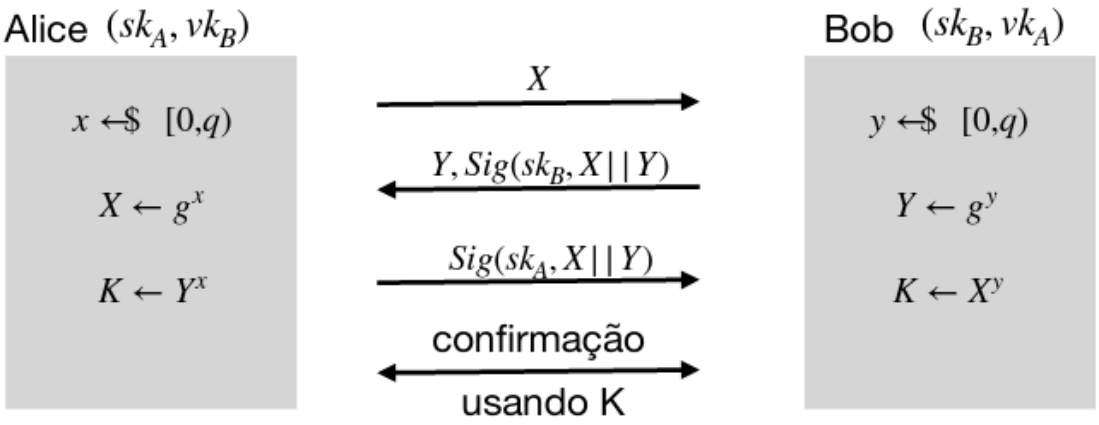
Como não há identificação, podemos aplicar aqui o ataque do **man in the middle**.



Portanto, nunca podemos confiar numa chave pública sem identificação. Sempre podemos assumir que esta chave pode ser mudada por um atacante.

Avoid Man-In-The-Middle

Alice e Bob precisam conhecer de alguma forma a chave de verificação de um do outro.



É importante que assinemos os pares X e Y.

- Alice verifica a assinatura de Bob na segunda mensagem e só prossegue no protocolo se Bob de facto recebeu o mesmo X que ela enviou;
- A Alice devolve a confirmação;
- Só depois o Bob confirma que a ação foi concluída.

> Assumindo que isto é o que usamos no TLS, isto implica que os browsers estariam a fazer assinaturas. Mas não estão, porque temos um servidor para muitos clientes, então o browser (Alice) quando recebe uma mensagem do Bob (o servidor) por HTTPS tem a certeza de que está a falar com o servidor. O servidor não precisa saber com quem se fala, mas se autentica por meio de logins, etc.

Perguntas

1) No paradígma híbrido, devemos encriptar e assinar ou o contrário? Por que?

▼ Resposta

Devemos fazer o sign-then-encrypt. Porque se fizermos encrypt-then-sign a pessoa que assinou determinado conteúdo pode alegar que não sabia o que estava nele, já que estava encriptado. Desta forma perde-se a propriedade de não repúdio.

2) O que é perfect forward secrecy? Por que garantir isto é importante?

▼ Resposta

É um sistema em que a chave de encriptação é automaticamente modificada a cada encriptação. Desta forma se uma chave for comprometida garantimos que as informações anteriores não são igualmente expostas. Por exemplo, um hacker pode escutar a rede e guardar o histórico de mensagens de um chat. Quando uma chave é comprometida o histórico não é revelado.

3) Explique Diffie Hellman.

▼ Resposta

Protocolo vulnerável ao man-in-the-middle

Alice (A) tem um um número primo g (público) e eleva este número a um fator x**.** Assim, $g^x = X$. Quando (A) quiser uma comunicação com **Bob** (B), (A) irá enviar X (que muda a cada sessão). (B), por sua vez irá calcular $g^y = Y$, irá enviar Y para (A) e irá computar $X^y = g^{xy}$. (A) por sua vez irá calcular $Y^x = g^{xy}$ e este novo g^{xy} irá ser utilizado como chave.

O problema deste protocolo é que este é vulnerável ao man-in-the-middle.

Protocolo seguro

Alice (A) irá enviar X como anteriormente referido. No entanto a resposta de (B) irá conter uma assinatura. $Y, sig(sk_b, Y)$ e depois se (A) confirmar os dados enviará uma resposta $X, sig(sk_a, X)$. Se os dados de (A) se confirmarem a conexão é estabelecida.

4) Como Diffie-Hellman pode evitar o ataque de man-in-the-middle?

▼ Resposta

Pode ser evitado com o uso de assinaturas digitais.