Criprografia Assimétrica

Criptografia

António Santos

ISTEC

Tópicos

Comparação

Simétrica - chave utilizada para encriptar também é usada para desencriptar (chave privada).
Assimétrica trabalha com duas chaves distintas uma utilizada para encriptar, que é então chamada de chave pública, e outra para desencriptar, que é chamada de chave privada.

Desvantagens

- Maior tempo de processo nas mesmas condições em relação à chave simétrica.
- Chaves maiores que em sistemas simétricos.
- A mensagem criptografada é maior que a mensagem original.

Terminologia Relacionada à Criptografia Assimétrica

Chaves Assimétricas - Duas chaves relacionadas, uma pública e uma privada, que são usadas para realizar operações complementares, como encriptação e desencriptação ou geração e verificação de assinatura.
Certificado de Chave Pública- Um documento emitido e assinado digitalmente pela chave privada de uma autoridade de Certificação, que vincula o nome de um assinante a uma chave pública. O certificado indica que o assinante identificado tem o único controle e acesso à chave privada correspondente.

Terminologia Relacionada à Criptografia Assimétrica

Algoritmo Criptográfico de Chave Pública (Assimétrica)- Um algoritmo criptográfico que usa duas chaves relacionadas, uma pública e uma privada. As duas têm a propriedade de ser computacionalmente inviável derivar a chave privada a partir da pública.

Infraestrutura de Chave Pública (PKI) - Um conjunto de políticas, processos, plataformas de servidor, software e estações de trabalho usadas para fins de administrar certificados e pares de chave pública/privada, incluindo a capacidade de emitir, manter e revogar certificados de chave pública.

Algumas notas

Os algoritmos de chave pública são baseados em funções matemáticas, em vez de substituição e permutação
Criptografia de chave pública é assimétrica, envolvendo o uso de duas chaves separadas
Não mais segura contra criptoanálise do que a criptografia simétrica
Não veio substittuir a criptografia simétrica.

Criptossistemas de Chave Pública

- É computacionalmente inviável determinar a chave de decriptação dado apenas o conhecimento do algoritmo de criptografia e da chave de encriptação.

Além disso, alguns algoritmos, como RSA, também exibem esta característica:

- Qualquer uma das duas chaves relacionadas pode ser usada para encriptação, com a outra para a desencriptação.

Esquema de encriptação de chave pública

- Texto claro: essa é a mensagem ou dados legíveis que são alimentados no algoritmo como entrada.
- Algoritmo de encriptação: realiza várias transformações no texto claro.
- Chaves pública e privada: esse é um par de chaves que foi selecionado de modo que, se uma for usada para encriptação, a outra é usada para desencriptação. As transformações exatas realizadas pelo algoritmo dependem da chave pública ou privada que é fornecida como entrada.

Criptossistemas de Chave Pública

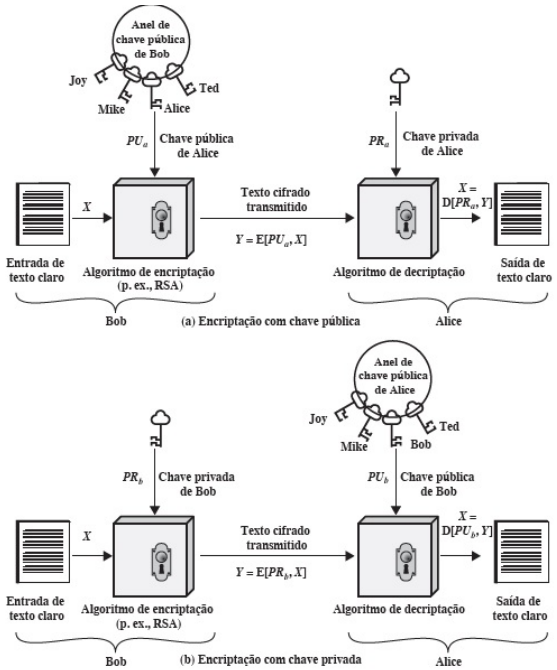

Figura: Criptografia de chave pública.

Esquema de encriptação de chave pública

- Texto cifrado: essa é a mensagem embaralhada produzida como saída. Ela depende do texto claro e da chave. Para determinada mensagem, duas chaves diferentes produzirão dois textos cifrados diferentes.
- Algoritmo de desencriptação: aceita o texto cifrado e a chave correspondente e produz o texto claro original.

As etapas essenciais

1. Cada utilizador gera um par de chaves a ser usado para a encriptação e desencriptação das mensagens.
2. Cada utilizador coloca uma das duas chaves num registrador público a sua chave publica. A chave parceira permanece privada. Cada utilizador mantém uma coleção de chaves públicas de outros.
3. Se Bob deseja enviar uma mensagem confidencial para Alice, ele encripta-a usando a chave pública de Alice.
4. Quando Alice recebe a mensagem, decripta-a usando a sua chave privada. Nenhum outro destinatário pode desencriptar a mensagem, pois somente Alice conhece a sua chave privada.

Encriptação Convencional e de Chave Pública.

Encriptação Convencional	Encriptação Chave Pública
Necessário para funcionar: 1. O mesmo algoritmo com a mesma chave é usado para encriptação e desencriptação.	Necessário para funcionar: 1. Um algoritmo é usado para encriptação, e um relacionado, para desencriptação com um par de chaves, uma para encriptação e outra para desencriptação.
2. O emissor e o receptor precisam compartilhar o algoritmo e a chave.	2. O emissor e o receptor precisam ter, cada um, uma chave do par (não a mesma).

Encriptação Convencional e de Chave Pública.

Necessário para a segurança:	Necessário para a segurança:
1. A chave precisa permanecer secreta.	1. Uma das duas chaves precisa permanecer secreta.
2. Deverá ser impossível, ou pelo menos impraticável, decifrar uma mensagem se a chave for mantida secreta.	2. Deverá ser impossível decifrar uma mensagem se uma das chaves for mantida secreta.
3. Conhecer o algoritmo e amostras do texto cifrado devem ser insuficientes para determinar a chave.	3. Conhecer o algoritmo, uma das chaves e amostras do texto cifrado devem ser insuficientes para saber a outra chave.

Encriptação chave Pública

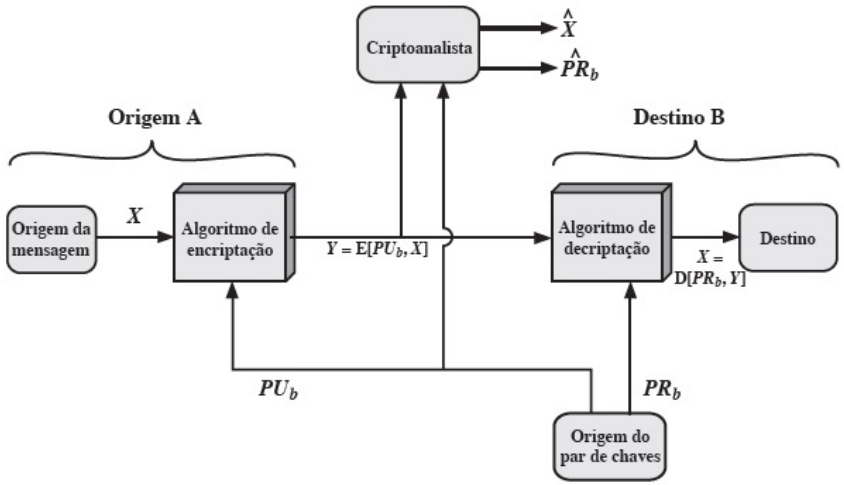

Figura: Criptossistema de chave pública: sigilo.

Encriptação chave Pública

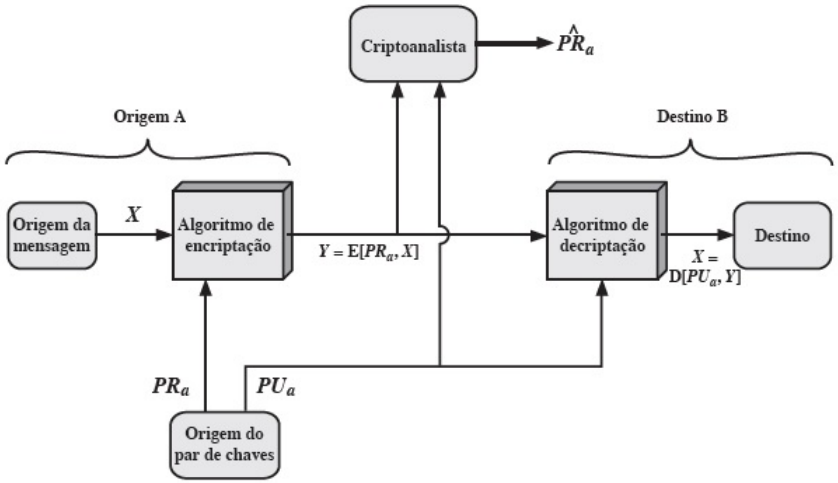

Figura: Criptossistema de chave pública: autenticação.

Encriptação chave Pública

Com a mensagem X e a chave de encriptação PU_b como entrada, A forma o texto cifrado $Y = [Y_1, Y_2, \cdots, Y_N]$:

$$Y = E(PU_b, X)$$

O receptor mal intencionado, de posse da chave privada correspondente, é capaz de inverter a transformação:

$$X = D(PR_b, Y)$$

Encriptação chave Pública

Encriptação de chave pública para oferecer autenticação:

$$Y = E(PR_a, X)$$

$$X = D(PU_a, Y)$$

Encriptação chave Pública

É possível oferecer a função de autenticação e confidencialidade com um uso duplo do esquema de chave pública:

$$Z = E(PU_b, E(PR_a, X))$$

$$X = D(PU_a, D(PR_b, Z))$$

Encriptação chave Pública

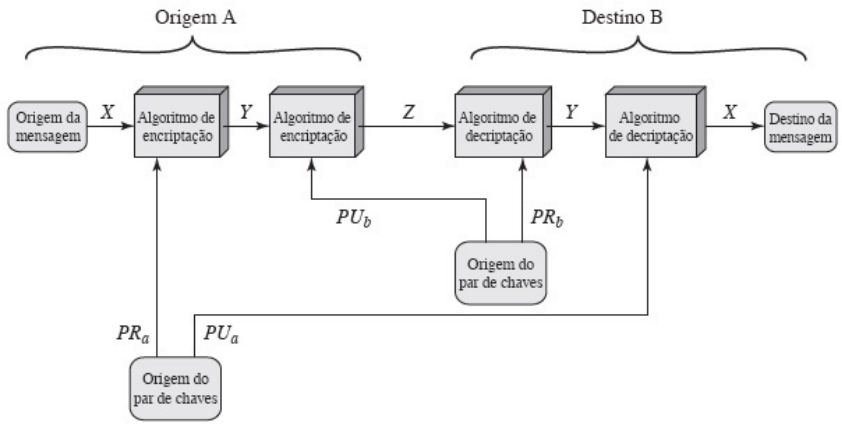

Figura: Criptossistema de chave pública: autenticação e sigilo.

Aplicações

1. Encriptação/desencriptação: o emissor encripta uma mensagem com a chave pública do destinatário.
2. Assinatura digital: o emissor "assina"uma mensagem com sua chave privada. A assinatura é feita por um algoritmo criptográfico aplicado à mensagem ou a um pequeno bloco de dados que é uma função da mensagem.
3. Troca de chave: dois lados cooperam para trocar uma chave de sessão. Várias técnicas diferentes são possíveis, envolvendo a(s) chave(s) privada(s) de uma ou de ambas as partes.

Aplicações

Algoritmo	Encriptação / Desencriptação	Assinatura Digital	Troca de Chave
RSA	Sim	Sim	Sim
Curva Elíptica	Sim	Sim	Sim
Diffie-Hellman	Não	Não	Sim
DSS	Não	Sim	Não

Requisitos para Criptografia de chave pública

1. É computacionalmente fácil para uma parte B gerar um par (chave pública PU_b, chave privada PR_b).
2. É computacionalmente fácil que um emissor A, conhecendo a chave pública e a mensagem a ser encriptada, M, gere o texto encriptado:

$$C = E(PU_b, M)$$

3. É computacionalmente fácil que o receptor B desencripte o texto cifrado resultante usando a chave privada para recuperar a mensagem original:

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$$

Requisitos para Criptografia de chave pública

1. É computacionalmente inviável que um invasor, conhecendo a chave pública, PU_b, determine a chave privada, PR_b.
2. É computacionalmente inviável que um invasor, conhecendo a chave pública, PU_b, e um texto cifrado, C, recupere a mensagem original, M.
   Podemos incluir um sexto requisito que, embora útil, não é necessário para todas as aplicações de chave pública:
3. As duas chaves podem ser aplicadas em qualquer ordem:

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$$

Requisitos para Criptografia de chave pública

Uma função de mão única com alçapão é uma família de funções reversíveis f_k, tal que

$Y = f_k(X)$ fácil, se k e X forem conhecidos

$X = f_k^{-1}(Y)$ fácil, se k e Y forem conhecidos

$X = f_k^{-1}(Y)$ inviável, se Y for conhecido, mas k não

Definição

Em 1977, os professores do MIT, Ronald L. Rivest, Adi Shamir e Leonard M. Adleman apresentaram o algoritmo assimétrico RSA, baseado em teorias clássicas dos números. O RSA envolve um par de chaves, a chave pública, que pode ser conhecida por todos e a chave privada, que deve ser mantida em segredo, pois toda mensagem cifrada usando uma chave pública pode ser decifrada, (única e exclusivamente), pela chave privada.
A criptografia RSA é tida como uma das mais seguras e atua diretamente na Internet, em e-mails e compras on-line.

Método

Passo 1: Escolhe-se aleatoriamente 2 números primos (p e q), entre 512 e 2048 bits. Números com 2048 bits possuem 617 dígitos no sistema decimal e é a recomendação da RSA Data Security.
Passo 2: Calcula–se $n = pq$.
Passo 3: Calcula–se a função totient (função ϕ de Euler): $\phi(n) = (p-1)(q-1)$.
Passo 4: Escolhe-se um inteiro d, de modo aleatório, tal que $1 < d < \phi(n)$ de forma que d, e $\phi(n)$ sejam primos entre si.
Passo 5: Calcula-se um número $e \in \mathbb{Z}^*_{\phi(n)}$ de forma que $e \times d = 1$. Ou em outras palavras, um e que seja o inverso multiplicativo de d em $\mathbb{Z}^*_{\phi(n)}$.

Método

A chave pública: o par de números n e d.
A chave privada: o par de números n e e.
Os valores p e q devem ser mantidos em segredo ou destruídos.
Para cifrar uma mensagem com esse algoritmo é realizado o seguinte cálculo:
$C(M) = M^d \ \ modn$,
onde $C(M)$ é a mensagem cifrada, M é o texto original, d e n são dados a partir da chave pública (n, d). Única chave que pode decifrar a mensagem $C(M)$ é a chave privada (n, e) através do cálculo de: $T(C) = C^e \ mod \ n$.

Exemplo

Bob e Alice desejam criar um forma segura de comunicação.
Assim Bob começa escolhendo dois números primos:
$p = 1231$ e $q = 337$.
Estes números podem ser obtidos através do Crivo de Erastótenes.
Depois calcula-se o produto:
$n = pq = 1231 * 337 = 414847$.

Exemplo

$$\phi(n) = (p-1)(q-1)$$
$$\phi(414847) = (1231-1)(337-1)$$
$$\phi(414847) = 413280$$

Exemplo

$mdc(413280, 2) = 2$
$mdc(413280, 3) = 3$
...
$mdc(413280, 9) = 9$
$mdc(413280, 10) = 10$
$mdc(413280, 11) = 1$
...
$mdc(413280, 211243) = 1$
Assim podemos tomar $d = 211243$, embora o 11 e muitos outros valores possam substituí-lo.

Exemplo

Como $mdc(\phi, d) = 1$ então como consequência do Teorema de Bézout $\phi(n)$ e d podem ser escritos como combinação linear.

$$d * e - \phi(n) * r = 1$$

$$211243 * e - 413280 * r = 1$$

Com $r, e \in \mathbb{Z}$.
Para calcular r e e Bob pode usar o Algoritmo de Euclides estendido. Que dará como resultado $r = -84924$ e $e = 166147$.

Exemplo

e é de fato o inverso multiplicativo de d em $\mathbb{Z}_{413280}$, pois:

$$e * d = \overline{166147} * \overline{211243} = \overline{35097390721} = \overline{1}$$

Exemplo

Finalmente as funções de criptografia e descriptografia podem ser criadas:
$C(M) = M^{211243} \ mod \ 414847$ (Criptográfica)
$T(C) = C^{166147} \ mod \ 414847$ (Descriptográfica)

Exemplo

Seja $M = 224455$ então:

$$C(224455) = 224455^{21243} \; mod \; 414847 = 376682.$$

O valor 376682 pode então ser transmitida a Bob que isoladamente a decodifica usando sua chave (chave privada):

$$T(376682) = 376682^{166147} \; mod \; 414847 = 224455.$$

Vulnerabilidade do RSA

Por ser um sistema de criptografia muito utilizado, o RSA tem tido algumas das suas vulnerabilidades pesquisadas e analisadas. Uma primeira abordagem de um ataque ao RSA teria como objetivo a chave pública por meio da fatorização de n para se chegar aos primos p e q. Este é um exemplo de ataque de força bruta ao RSA. Contudo apesar da melhoria constante dos algoritmos de fatorização de números inteiros e da capacidade de processamento dos computadores ao longo dos anos, este método é uma ameaça considerada distante da realidade.

Problema

O maior problema para a implementação do RSA está no fato de que este necessita de números grandes para oferecer uma segurança razoável. E é um fato bem conhecido que muitas linguagens de programação como Pascal e Fortran não conseguem lidar com números de 20 ou 30 dígitos de maneira trivial. Também as estratégias encontradas para contornar este problema, como a álgebra simbólica.

Exemplo

Durante a explicação do DES utilizamos o bloco M de bits para encriptação:

$M =$

$0001001100110100010101110111100110011011101111001101111111100$

Exemplo

Primeiro convertemos esse bloco para decimal.

$00010011 = 19$

$00110100 = 52$

$01010111 = 87$

$01111001 = 121$

$10011011 = 155$

$10111100 = 188$

$11011111 = 223$

$11110001 = 241$

Para encriptação escolhemos $p = 17$ e $q = 41$ de modo a termos $e = 229$ e $d = 109$.

Exemplo

Quebrar o bloco M em sub blocos onde cada sub-bloco deve ter um valor inferior a n, que neste caso é igual a 697.
$M = 196287121155188223241$

Exemplo

Texto	Resultado
$196^{109}\ mod(697)$	688
$287^{109}\ mod(697)$	410
$121^{109}\ mod(697)$	185
$155^{109}\ mod(697)$	032
$188^{109}\ mod(697)$	477
$223^{109}\ mod(697)$	508
$241^{109}\ mod(697)$	607

Exemplo

Obtendo como mensagem cifrada o bloco
$M' = 688410185032477508607$.
Que em binário seria:
$M' =$
$1010110000011001101000101110010000100000011101110101111111001$0

Exemplo

Texto	Resultado
$688^{109}\ mod(697)$	196
$410^{109}\ mod(697)$	287
$185^{109}\ mod(697)$	121
$032^{109}\ mod(697)$	155
$477^{109}\ mod(697)$	188
$508^{109}\ mod(697)$	223
$607^{109}\ mod(697)$	241

Tabela de codificação de caracteres para criptografia RSA

A=10	B=11	C=13	D=14	E=15
F=16	G=18	H=17	I=20	J=21
K=22	L=23	M=24	N=25	O=26
P=27	Q=28	R=29	S=30	T=31
U=32	V=33	W=34	X=35	Y=36
Z=37	0=38	1=39	2=40	3=41
4=42	5=43	6=44	7=45	8=46
9=47		Espaço = 99		

Exemplo

"TCC de Criptografia RSA e Algoritmo AKS 2015", após a codificação com a tabela anterior obtemos
3113139914159913292027312618291016201099293010991023182

Exemplo

Separamos o número obtido em blocos menores que nosso n,
que é 19549 (chave):
3113 - 13991 - 4159 - 9132 - 9202 - 7312 - 6182 - 9101 - 620 - 10992 - 930

Exemplo

Se $(b_i)^c \equiv C_i (mod\ n)$:

$b_1 \rightarrow 3113^{199} \equiv 16261\ (mod\ 19549) \rightarrow C_1$

$b_2 \rightarrow 13991^{199} \equiv 640\ (mod\ 19549) \rightarrow C_2$

$b_3 \rightarrow 4159^{199} \equiv 13360\ (mod\ 19549) \rightarrow C_3$

...

$b_{19} \rightarrow 9940^{199} \equiv 11688\ (mod\ 19549) \rightarrow C_{19}$

$b_{20} \rightarrow 3839^{199} \equiv 10669\ (mod\ 19549) \rightarrow C_{20}$

$b_{21} \rightarrow 43^{199} \equiv 788\ (mod\ 19549) \rightarrow C_{21}$

Exemplo

$(199, 19549)$

16261 - 640 - 13360 - 2561 - 6344 - 18577 - 11006 - 6816 - 7365 - 945 - 41

Exemplo

Para descodificar uma mensagem é necessária uma chave de descodificação

$$c * d + \beta\phi(n) = 1.$$

No caso descrito:
$199 * d + \beta\phi(19549) = 1.$

Exemplo

Para este n resulta 113 e 173, logo temos que
$\phi(19549) = (113-1)(173-1) = 19264$.
$199 * d + \beta * 19264 = 1$.
Portanto:
$199 * 14327 + (-148) * 19264 = 1$. $d = 14327$, e a chave de descodificação é $(14327, 19549)$

Exemplo

$$(C_i)^d \equiv D_i(mod\ n)$$

Para codificamos e descodificamos uma mensagem, precisamos então mostrar que $b_i = D_i$. Durante o processo fizemos o seguinte:

$$((b_i)^c)^d \equiv (C_i)^d \equiv D_i(mod\ n).$$

E sabemos que:

$$(b_i)^{c*d} \equiv b_i^{1-\beta*\phi(n)} \equiv D_i(mod\ n).$$

Exemplo

Por fim:

$$b_i * ((b_i)^{\phi(n)})^{-\beta} \equiv D_i(mod\ n).$$

Pelo teorema de Euler $(b_i)^{\phi(n)} \equiv 1(mod\ n)$, concluímos que:

$$b_i * (1)^{-\beta} \equiv b_i = D_i(mod\ n).$$

Conforme o nosso caso numérico, temos, para o bloco b_1, que: $((5115)^{199})^{14327} \equiv (16261)^{14327} \equiv 3113(mod\ 19549)$.

Conceitos

A teoria das curvas elípticas baseia-se na Geometria algébrica, quando definidas sobre corpos finitos, tem encontrado diversas aplicações em Criptografia.
Corpos finitos fornecem uma quantidade inesgotável de grupos abelianos que, mesmo quando se tem um número grande de elementos, ainda são adequados aos processos computacionais por causa da sua rica estrutura algébrica.
O estudo da criptografia de curvas elípticas é similar em vários aspectos ao estudo da criptografia no grupo multiplicativo de um corpo finito, mas existe uma maior flexibilidade para a escolha de um grupo associado a uma curva elíptica do que a um grupo multiplicativo de um corpo finito.

Definição

Seja K um corpo, com característica diferente de $2, 3$ e seja $X^3 + aX + b$ (onde $a, b \in K$), um polinómio cúbico sem raízes múltiplas. Uma curva elíptica sobre K é o conjunto de pontos $(x, y) \in K^2$ que satisfazem equação:

$$y^2 = x^3 + ax + b \tag{1}$$

juntamente com um elemento denotado O chamado ponto no infinito.

Conceitos

Se K é um corpo de característica 2, então uma curva elíptica sobre K num conjunto de pontos que satisfazem uma equação do tipo

$$y^2 + cy = x^3 + ax + b \tag{2}$$

ou então

$$y^2 + xy = x^3 + ax^2 + b \tag{3}$$

juntamente com um ponto no infinito O.

Conceitos

Se K é um corpo de característica 3, então a curva elíptica sobre K é um conjunto de pontos que satisfaz a equação

$$y^2 = x^3 + ax^2 + bx + c \tag{4}$$

Observação

Existe uma forma geral da equação de uma curva elíptica, conhecida como *Equação de Weierstrass*, válida para qualquer corpo:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{5}$$

Observação

Quando a característica é diferente de 2, podemos simplificar a equação completando quadrados e substituindo y por $\frac{1}{2}(y - a_1x - a_3)$. E então a forma geral pode ser transformada em

$$y^2 = 4x^3 + b_2x^2 + 2b_4x + b_6 \tag{6}$$

onde: $b_2 = a_1^2 + 4a_2$, $b_4 = 2a_4 + a_1a_3$ e $b_6 = a_3^2 + 4a_6$
e substituindo y por $2y$, tem-se:

$$y^2 = x^3 + ax^2 + bx + c \tag{7}$$

Observação

Seja $F(x,y) = 0$ uma equação implícita que define uma curva elíptica e que tem y e x como variáveis dessa função em (1) (ou (2 e 3), ou ainda,(4)), isto é, $F(x,y) = y^2 - x^3 - ax - b$, $F(x,y) = y^2 + cy + x^3 + ax + b$, $F(x,y) = y^2 + xy + x^3 + ax + b$, $F(x,y) = y^2 - x^3 - ax^2 - bx - c$, então um ponto (x,y) na curva é chamado "não-singular"ou ponto suave se pelo menos uma das derivadas parciais $\frac{\partial F}{\partial x}$ ou $\frac{\partial F}{\partial y}$ é não-nula neste ponto.

Exemplo

Considere a equação $Q = kP$, onde $Q, P \in E_p(a, b)$ e $k < p$.
Considere o grupo $E_{23}(9, 17)$. Esse é o grupo definido pela equação $y^2 \ mod \ 23 = (x^3 + 9x + 17) mod \ 23$. Qual é o logaritmo discreto k de $Q = (4, 5)$ à base $P = (16, 5)$?
O método da força bruta é calcular múltiplos de P até que Q seja encontrado. Assim, $P = (16, 5)$; $2P = (20, 20)$;
$3P = (14, 14)$; $4P = (19, 20)$; $5P = (13, 10)$; $6P = (7, 3)$;
$7P = (8, 7)$; $8P = (12, 17)$; $9P = (4, 5)$.

Troca de chaves

A troca de chaves usando curvas elípticas pode ser feita da maneira a seguir. Primeiro, escolha um inteiro grande q, que é um número primo p ou um inteiro na forma 2m.
$y^2\ mod\ p = (x^3 + ax + b) mod\ p$ ou a Equação
$y^2 + xy = x^3 + ax^2 + b$.
Isso define o grupo elíptico de pontos $E_q(a, b)$.
Selecione um ponto de base $G = (x_1, y_1)$ em $E_p(a, b)$ cuja ordem é um valor muito grande n. A ordem n de um ponto G numa curva elíptica é o menor inteiro positivo n : $nG = 0$ e G são parâmetros do criptossistema conhecido por todos os participantes.

Troca de chaves

1. A seleciona um inteiro n_A menor que n. Essa é a chave privada de A. A, então, gera uma chave pública $P_A = n_A \times G$, a chave pública é um ponto em $E_q(a, b)$.
2. B, de modo semelhante, seleciona uma chave privada n_B e calcula uma chave pública P_B.
3. A gera a chave secreta $k = n_A \times P_B$. Já B desencadeia a chave secreta $k = n_B \times P_A$.

Elementos Públicos Globais

$E_q(a, b)$ CE com elementos a, b e q: q é um primo ou 2^m

G ponto na curva elíptica cuja ordem é o valor grande n

Geração de Chave do Utilizador A

Selecione Privada n_A $n_A < n$

Calcule Pública P_A $P_A = n_A \times G$

Geração de Chave do Utilizador B

Selecione Privada n_B $n_B < n$

Calcule Pública P_B $P_B = n_B \times G$

Cálculo da Chave Secreta pelo Utilizador A

$K = n_A \times P_B$

Cálculo da Chave Secreta pelo Utilizador B

$K = n_B \times P_A$

Troca de chaves

Os dois cálculos na etapa 3 produzem o mesmo resultado, porque

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A$$

Troca de chaves

Para quebrar esse esquema, um intruso teria que ser capaz de calcular k, dados G e k_G, o que é considerado difícil.
EX: Seja $p = 211$; $E_p(0, -4)$, que é equivalente à curva $y^2 = x^3 - 4$; e $G = (2, 2)$. Pode-se calcular que $240G = O$.
A chave privada de A é $n_A = 121$, de modo que a chave pública de A é $P_A = 121(2, 2) = (115, 48)$.
A chave privada de B é $n_B = 203$, de modo que a chave pública de B é $203(2, 3) = (130, 203)$. A chave secreta partilhada é $121(130, 203) = 203(115, 48) = (161, 69)$.

Pontos (diferentes de O) na curva elíptica $E_{23}(1, 1)$

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

Encriptação

Cada utilizador A seleciona uma chave privada nA e gera uma chave pública $P_A = n_A \times G$.
Para encriptar e enviar uma mensagem P_m a B, A escolhe um inteiro positivo aleatório k e produz o texto encriptado C_m consistindo no par de pontos:

$$C_m = \{k_G, P_m + kP_B\}$$

Observe que A usou a chave pública de B, P_B.
Para desencriptar: B multiplica o primeiro ponto no par pela chave privada de B e subtrai o resultado do segundo ponto:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$$

Exemplo

Os elementos públicos globais são $q = 257$;
$E_q(a, b) = E_{257}(0, -4)$, que é equivalente à curva $y^2 = x^3 - 4$; e
$G = (2, 2)$.
A chave privada de Bob é $n_B = 101$, a sua chave pública é
$P_B = n_B G = 101(2, 2) = (197, 167)$.
Alice deseja enviar uma mensagem a Bob, que está codificada no ponto elíptico $P_m = (112, 26)$.

Exemplo

Alice escolhe o inteiro aleatório $k = 41$ e calcula $kG = 41(2, 2) = (136, 128)$, $kP_B = 41(197, 167) = (68, 84)$ e $P_m + kP_B = (112, 26) + (68, 84) = (246, 174)$. Alice envia o texto cifrado $C_m = (C_1, C_2) = \{(136, 128), (246, 174)\}$ a Bob.
Bob recebe o texto cifrado e calcula $C_2 - n_B C_1 = (246, 174) - 101(136, 128) = (246, 174) - (68, 84) = (112, 26)$.

Segurança

Algoritmos de chave simétrica	Algoritmo Diffie-Hellman, assinatura digital	RSA (tamanho de n em bits)	ECC (tamanho do módulo em bits)
80	L = 1024 N = 160	1024	160–223
112	L = 2048 N = 224	2048	224–255
128	L = 3072 N = 256	3072	256–383
192	L = 7680 N = 384	7680	384–511
256	L = 15360 N = 512	15360	512+

L = tamanho da chave pública, N = tamanho da chave privada

Definição

Uma função de hash aceita uma mensagem de tamanho variável M como entrada e produz um valor de hash de tamanho fixo $h = H(M)$.

Propriedades

1. Para a mesma função hash, qualquer que seja o comprimento da entrada M, a saída $h = H(M)$ deve ter um tamanho fixo.
2. Para cada M, $h = H(M)$ deve ser único.
3. Tem que ser rápido e fácil de calcular.
4. Não se pode retornar a M de h.

Propriedades

1. Não pode apresentar colisões. Isso significa que, para dois A diferentes, o mesmo B. não pode ser dado. Observando as propriedades 1 e 2, vemos que isso é impossível. A função MD5 resulta num hash de 128 bits. Isso significa que, no máximo, existem apenas 2^{128} textos diferentes, o que é falso. Portanto, é muito importante que as colisões sejam mínimas e que encontrá-las seja muito difícil.

Funções

Funções de hash criptográfico, é necessário de forma adicional que sejam uniformes (para um A escolhido aleatoriamente, todos os valores de hash são equiprovaveis) e com um efeito de avalanche (a mudança de um único bit em A assume um B completamente diferente). Podemos distinguir dois grupos de funções: aqueles que visam manter a integridade das mensagens (deteção de modificação) e os que se destinam a verificar a origem da mensagem (autenticação).

Descrição

As funções de hash são funções que recebem uma entrada de comprimento indeterminado e produzem um valor de comprimento fixo. Mesmo as funções mais simples de hash têm muita aplicabilidade. As tabelas de hash, uma estrutura de dados comum, dependem das funções de hash. Estas funções simples de hash realmente apenas garantem uma coisa: para duas entradas idênticas, elas produzirão uma saída idêntica. É importante ressaltar que não há garantia de que duas saídas idênticas impliquem que as entradas sejam iguais.

Descrição

Para uma função hash criptográfica, queremos que seja impossivelmente difícil:

1. Modificar uma mensagem sem alterar o hash.
2. Gerir uma mensagem que tenha um determinado hash.
3. Encontrar duas mensagens diferentes com o mesmo hash.

Introdução

O MD5 é algoritmo de redução criptográfica de 128 bits ampliamente usado. É baseado numa função de hash projetada por Ronald Rivest em 1991 como uma extensão do MD4. Essa função hash gera resumos de 128 bits. Em 1993, Bert den Boer e Antoon Bosselaers publicaram um artigo demonstrando "pseudo-colisões"para a função de compressão do MD5. Dobbertin expandiu essa pesquisa e conseguiu produzir colisões para a função de compressão. Em 2004, com base no trabalho de Dobbertin, Xiaoyun Wang, Dengguo Feng, Xuejia Lai e Hongbo Yu mostraram que o MD5 é vulnerável a ataques de colisão reais.

Etapas

- Adição de bits. A mensagem é estendida de forma a uma longitude menor que 448 que seja múltiplo de 512. Este passo é realizado imediatamente após a longitude seja congruente com 448 módulo 512.
- Comprimento da mensagem. Um número inteiro de 64 bits que representa o comprimento da mensagem antes da adição ser concatenada no final do resultado da etapa anterior. Se o tamanho da mensagem for superior a 64 bits, apenas os primeiros 64 bits serão usados. Após esta etapa, a mensagem é um múltiplo exato de 512. Por sua vez, a mensagem é um múltiplo de 16.

Etapas

- Inicie o buffer MD. Um buffer de quatro palavras começa com certos valores. Cada palavra A, B, C e D tem 32 bits. As palavras têm os seguintes valores:
  A: 01 23 45 67
  B: 89 ab cd ef
  C: fe dc ba 98
  D: 76 54 32 10

Etapas

1. Processado a mensagem em blocos de 16 palavras. Tomamos quatro funções auxiliares cuja entrada é três palavras e sua saída é uma:

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$

$$H(B,C,D) = B \otimes C \otimes D$$

$$I(B,C,D) = C \otimes (B \vee \neg D)$$

   tal que $\otimes, \vee, \wedge$ e $\neg$ representam: XOR, AND, OR e NOT.
2. Após executar 16 ciclos nos quais uma série de operações é realizada com as funções anteriores em B, C e D, obtemos a saída $ABCD$.

Descrição

O algoritmo SHA-1 foi desenvolvido pela NSA, para ser incluído no padrão DSS (Digital Signature Standard). Diferentemente dos algoritmos de criptografia propostos por essa organização, o SHA-1 é considerado seguro e livre de backdoors, pois o fato de o algoritmo ser realmente seguro favorece os interesses da própria NSA. Produz assinaturas de 160 bits, a partir de blocos de 512 bits da mensagem original.

Descrição

O algoritmo é semelhante ao MD5, com a diferença de que usa a classificação big endian. Ele é inicializado da mesma maneira, ou seja, adicionando ao final da mensagem um seguido de quantos zeros forem necessários até completar 448 bits no último bloco, justapondo o comprimento em bits da própria mensagem (neste caso, o primeiro byte da sequência será o mais significativo).

Descrição

Diferentemente do MD5, o SHA-1 usa cinco registros de 32 bits em vez de quatro, que devem ser inicializados antes do processamento do primeiro bloco com os seguintes valores:

$A = 67452301$

$B = EFCDAB89$

$C = 98BADCFE$

$D = 10325476$

$E = C3D2E1F0$

Descrição

Depois que os cinco valores são inicializados, eles são copiados em cinco variáveis, a, b, c, d e e. O loop principal tem quatro rodadas com 20 operações cada:

$$F(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$

$$G(X, Y, Z) = X \otimes Y \otimes Z$$

$$H(X, Y, Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$$

Descrição

A operação F é usada na primeira rotação ($0 \leq t \leq 19$), G no segundo ($20 \leq t \leq 39$) e no quarto ($60 \leq t \leq 79$) e H no terceiro ($40 \leq t \leq 59$). Além disso, quatro constantes são usadas, uma para cada rotação:

$K_0 = 5A827999$

$K_1 = 6ED9EBA1$

$K_2 = 8F1BBCDC$

$K_3 = CA62C1D6$

Descrição

O bloco de mensagens m é dividido em 16 partes de 32 bits m_0 a m_{15} e convertido em 80 partes de 32 bits w_0 a w_{79} usando o seguinte algoritmo:

$w_t = mt$ Para $t = 0, \cdots, 15$

$w_t = (w_{t-3} \otimes w_{t-8} \otimes w_{t-14} \otimes w_{t-16}) \triangleleft 1$ Para $t = 16, \cdots, 79$

Introdução

No final dos anos 1960 e início dos anos 1970, Stephen Wiesner publicou um artigo Codificação Conjugada (Conjugate Coding), no qual introduziu a primitiva chamada multiplexação (multiplexing), que foi o ponto de partida da criptografia quântica. Embora a sua primitiva fosse equivalente ao que foi mais tarde chamado $1-2$ Transferência Inconsciente.
A Criptografia quântica refere-se a técnicas de criptografia cuja segurança se deve a fenômenos quânticos, para tal utiliza os princípios da Mecânica Quântica para garantir uma comunicação segura. Com ela, emissor e receptor podem criar e partilhar uma chave secreta para encriptar e desencriptar as suas mensagens.

Introdução

A sua grande segurança está no princípio da incerteza de Heisenberg, que determina que é impossível termos certeza absoluta de determinados valores para sistemas quânticos, como posição e momento, ou seja, caso medirmos um valor para a posição de uma partícula, o seu valor no momento vai possuir incerteza.

Introdução

A criptografia quântica destaca-se face aos outros métodos criptográficos por não necessitar de comunicações secretas prévias, permitir a deteção de intrusos e ser segura mesmo que o intruso possua um poder computacional ilimitado. Na verdade, ela é totalmente segura, exceto nas situações em que o intruso consiga remover e inserir mensagens do canal de transmissão (poder ler e remover a mensagem, criar uma cópia e reenviá-la).

Transferência Inconsciente

Protocolo Transferência Inconsciente é um protocolo de troca de mensagens no qual alguém envia alguma informação para o receptor, mas permanece sem saber (oblivious) o que é recebido.
A primeira forma de OTP foi apresentada em 1981 (ela é baseada no RSA). Neste processo, o transmissor envia a mensagem para o receptor com probabilidade $0,5$, e permanece sem saber se o receptor recebeu a mensagem.
Uma outra forma de transferência denominada transferência inconsciente $1-2$ ou (1 out of 2 oblivious transfer) foi desenvolvida com a finalidade de construir protocolos para computação multiparticipante segura.

Transferência Inconsciente de Rabin

Transferência Inconsciente de Rabin, o emissor gera um módulo público RSA $N = pq$, em que p e q são enormes números primos, e um expoente e é relativamente primo a $(p-1)(q-1)$. O emissor encripta a mensagem m como $m^e mod N$.
O emissor envia N, e e $m^e mod N$ para o receptor.
O receptor pega um x aleatório módulo N (x aleatório modulo N) e envia $x^2 mod N$ para o emissor (O $MDC((x, N) = 1$, com enorme probabilidade, o que assegura que há quatro raízes quadradas de $x^2 mod N$).

Transferência Inconsciente $1-2$

O emissor gera chaves RSA, incluindo o módulo N, o expoente público e e o expoente privado d. Ele pega duas mensagens aleatórias x_0 e x_1 e envia N, e, x_0 e x_1 para o receptor.
O receptor pega uma mensagem aleatória k, encripta-a, soma xb a essa encriptação de k módulo N e envia o resultado para o emissor.

Transferência Inconsciente $1-n$ e Transferência Inconsciente $k-N$

Estes protocolos podem ser definidos como a generalização do Transferência Inconsciente $1-2$.
Especificamente, o emissor tem n mensagens, o receptor tem um índice i, e ele deseja receber a $i-$ésima mensagem entre as mensagens do emissor, sem que este conheça i. Quanto ao emissor, ele deseja se assegurar de que o receptor receba apenas uma das n mensagens.

Descrição

A chave criptografada surge quando Alice (o emissor) envia bits aleatórios 0 e 1 para Bob (o receptor), que decide (aleatoriamente) o modo como irá medir os bits.
Devido ao Princípio de Incerteza, Bob só pode medir os bits num dos modos (retilíneo ou diagonal), mas não em ambos os modos. Apenas os bits que forem medidos por Bob que estiverem no mesmo modo que foi enviado por Alice estarão na direção correta, e apresentarão o valor apropriado.

Descrição

Após a transmissão feita por Alice, Bob comunica-se com ela em canal aberto para informar qual dos dois modos usou para medir cada fóton (não revela o valor de cada bit). Alice, em seguida, informa a Bob quais dos modos foram medidos corretamente. Estes modos constituem a chave criptográfica que será utilizada entre Alice e Bob, para cifrar ou decifrar mensagens.

Descrição

Se a transmissão for interceptada (por Trudy), devido ao Princípio de Incerteza, Trudy não conseguirá medir os dois modos, e se reenviar os bits como ela os mediu de volta a Bob, inevitavelmente introduzirá erros. Alice e Bob podem detectar a intrusão pela comparação dos bits selecionados, para ver se contém erros.

Qubits

A unidade fundamental de informação é o assim chamado dígito binário, ou bit (contração de binary digit), um sistema de dois níveis lógicos distintos representado fisicamente por um sistema com também dois níveis físicos distintos, como dois valores de tensão elétrica. Analogamente, em comunicações quânticas a entidade fundamental é chamada de bit quântico, ou qubit (quantum bit).

Descrição

$$|\varphi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{8}$$

Descrição

$$|Y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{9}$$

Descrição

Esta girará o sistema em $45°$ no espaço de Hilbert, resultando em um estado bem definido - neste caso, o estado $H|Y\rangle = |0\rangle$.

$$H|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{10}$$

$$H|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Descrição

$$\Phi |0\rangle = e^{j\phi} |0\rangle \tag{11}$$
$$\Phi |1\rangle = |1\rangle$$

Definição

É possível demonstrar a impossibilidade de se efetuar uma cópia perfeita de um estado quântico desconhecido. Uma copiadora quântica ideal deveria produzir na saída o próprio estado de entrada e uma cópia, como na equação 12, sendo $|\Psi\rangle$ o estado original do qubit a ser copiado, $|b\rangle$ representa a cópia em branco que deverá assumir o estado desejado e $|0\rangle$ representa o estado inicial da copiadora quântica, pertencente a um espaço de Hilbert arbitrário.

Descrição

$$|\Psi\rangle \otimes |b\rangle \otimes |0\rangle \rightarrow |\Psi\rangle \otimes |\Psi\rangle \otimes |f_{\Psi}\rangle \tag{12}$$

Descrição

Assim, para dois estados de uma determinada base $|V\rangle$ e $|H\rangle$, representando, respectivamente, os estados de polarização vertical e horizontal de um fóton, ortogonais entre si, obtém-se as equação 13.

$$\begin{aligned} \Phi\,|V,b,0\rangle &\rightarrow |V,V,f_V\rangle \\ \Phi\,|H,b,0\rangle &\rightarrow |H,H,f_H\rangle \end{aligned} \tag{13}$$

Descrição

Porém, ao se tentar copiar estados superpostos, por exemplo, uma polarização linear com ângulo de $+45°$, representada por $|+45°\rangle$, o resultado, dado pela equação 14, será divergente do apresentado nas equações 13.

$$\begin{aligned} \Phi\,|+45,b,0\rangle &= \frac{1}{\sqrt{2}}(|V\rangle + |H\rangle) \otimes |b,0\rangle \qquad (14) \\ &= \frac{1}{\sqrt{2}}(|V,V,f_V\rangle + |H,H,f_H\rangle) \neq |+45,+45,f_{+45}\rangle \end{aligned}$$

Descrição

Em certos casos, pode-se dizer que duas partículas estão emaranhadas se apresentarem correlação tal que, ao se efetuar a medição de uma delas, o estado quântico da outra pode ser predito. Um par emaranhado pode ser representado como uma superposição de estados produto, compondo um estado não-separáveis de duas (ou mais) partículas.

Descrição

$$|\Psi_{12}\rangle = \frac{1}{\sqrt{2}}(|0\rangle_1 |1\rangle_2 + e^{j\chi} |1\rangle_1 \otimes |0\rangle_2) \tag{15}$$

Paradoxo EPR

O paradoxo EPR pode ser explicado considerando-se uma fonte de pares de fótons emaranhados, sendo $|H\rangle$ e $|V\rangle$ os estados de polarização ortogonais componentes de uma base. Os fótons propagam-se a partir da fonte numa direção, cada um num sentido, podendo ser medidos nos extremos.

$$\left|\Phi^{+}\right\rangle_{12} = \frac{1}{\sqrt{2}}(|H\rangle_1 |V\rangle_2 + |V\rangle_1 \otimes |H\rangle_2) \tag{16}$$

Polarization beam-splitter

Polarization beam-splitter, com um detector de fótons únicos em cada um de seus dois braços. Estes podem, dependendo do estado de polarização do fóton incidente no PBS, acusar os bits 0 ou 1 em cada um. A equação abaixo representa a predição pela teoria quântica do número de vezes em que o ponto A obtém a medição de um bit 1 no seu sistema com orientação α e o ponto B obtém o bit 1 com orientação β, sendo $N : 0$ o número de pares emaranhados emitidos pela fonte.

$$N(1_\alpha, 1_\beta) = \frac{N_0}{2} \cos^2(\alpha - \beta) \tag{17}$$

Descrição

Um sistema de distribuição quântica de chaves constitui-se basicamente de um transmissor, geralmente denominado na literatura como Alice, um receptor, chamado de Bob, um canal quântico para a transmissão da chave quântica, um canal público para a reconciliação da chave, que pode ser monitorizado, mas não modificado (canal autenticado), e a possível presença de um espião, referido como Trudy, que deve ser considerado como capaz de qualquer tipo de interceptação.

Descrição

Inicialmente proposto por Bennett e Brassard em 1984, o protocolo $BB84$ utiliza quatro estados, que formam duas bases não-ortogonais em um espaço de Hilbert, para codificar a chave. Essas bases devem ser maximamente conjugadas, ou seja, qualquer par de vetores, um de cada base, deve apresentar a mesma superposição.

Descrição

Alice		Bob		
Base	Bit	Base	Determinação	Bit
α	0	α	Sim	0
		β	Não	0 ou 1
	1	α	Sim	1
		β	Não	0 ou 1
β	0	α	Não	0 ou 1
		β	Sim	0
	1	α	Não	0 ou 1
		β	Sim	1

Tabela: Possibilidades de combinações de bases para qubits codificados segundo o protocolo $BB84$.

Descrição

o protocolo B92 (criado por Bennett em 1992) são necessários apenas dois estados quânticos não ortogonais para que se possa proceder a distribuição quântica de chaves, e não quatro como no protocolo anterior.
Neste protocolo, Bob deverá proceder com uma medição POVM (positive operator valued measurement), ou seja, para cada base escolhida para medição, ele pode obter o resultado que determina o bit recebido (0 ou 1) ou pode não obter resultado $(-)$, ou seja, terá um sistema ternário com os bits 0 ou 1, caso a base esteja correta, ou a ausência de resultado, caso a base não esteja de acordo com a escolhida por Alice.

Descrição

Alice		Bob		
Base	Bit	Base	Resultado	Bit
α	0	α	Sim	0
		β	Não	$-$
β	1	α	Não	$-$
		β	Sim	1

Tabela: Possibilidades de combinações de bases para qubits codificados segundo o protocolo $B92$.

Descrição

A QKD também pode ser efetuada utilizando estados emaranhados. Deve haver uma fonte de pares de fótons emaranhados emitindo estados superpostos, assumidos como maximamente emaranhados, através de um canal quântico até Alice e Bob, recebendo, cada um, uma partícula na direção do eixo $\hat{Z}$, como na equação 18, onde $|V\rangle$ e $|H\rangle$ representam estados ortogonais de polarização.

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|V\rangle_1 |H\rangle_2 - |H\rangle_1 |V\rangle_2) \tag{18}$$

Probabilidade

$P \pm \pm(\Phi_n^a, \Phi_m^b)$ representa a probabilidade de ocorrer o resultado ± 1 com a base Φ_n^a e ± 1 com a base Φ_m^b.

$$E(\Phi_n^a, \Phi_m^b) = P_{++}(\Phi_n^a, \Phi_m^b) + P_{--}(\Phi_n^a, \Phi_m^b) - P_{+-}(\Phi_n^a, \Phi_m^b) - P_{-+}(\Phi_n^a, \Phi_m^b) \quad (19)$$

De acordo com as leis quânticas, a equação 19 resulta na equação 20.

$$E(\Phi_n^a, \Phi_m^b) = -\cos(\Phi_n^a - \Phi_m^b) \quad (20)$$

Descrição

Após a transmissão dos qubits e a reconciliação de bases, para que Alice e Bob compartilhem uma sequência de bits realmente idêntica, devem estimar a taxa de erro e corrigi-los. Cabe ressaltar que a taxa de erro de bits quânticos (QBER - *Qubit error rate*) se refere aos bits da sifted key, a chave provisória obtida após o anúncio público das bases. Tais erros devem-se a fatores como variações de polarização pela fibra, desestabilização de interferências ou filtragens imperfeitas, de acordo com o sistema em questão, e ainda à contagem de escuro dos detectores.

Probabilidade de ocorrência de um bit 1

Probabilidade de ocorrência de um bit 1 é dada por p_1.

$$H(X) = -p_0 \log_2^{p_0} -p_1 \log_2^{p_1} \tag{21}$$

Descrição

O sucesso da utilização da chave criptográfica requer que Alice e Bob tenham cópias idênticas. A correção de erro, próxima etapa de reconciliação, utiliza o canal público, e também deve ser o mais sigilosa possível.
Segundo o teorema da codificação de Shannon, o número mínimo de bits (r) que Alice e Bob devem compartilhar publicamente para corrigir os erros da chave (de comprimento n), sendo e a probabilidade de erro de transmissão de um bit, será dado pela equação 22.

$$r = n[-\epsilon \log_2 \epsilon - (1-\epsilon)\log_2(1-\epsilon)] \tag{22}$$