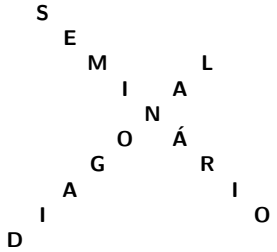


Criptologia; Contratos e Dinheiro Virtuais

Pedro Miguel Adão
4º ano da LMAC — Ciência da Computação
pad@math.ist.utl.pt

8 de Maio de 2001

Palavras Chave

sistema criptográfico, chave pública, chave privada, assinatura digital, dinheiro virtual.

Resumo

Quando queríamos guardar alguma coisa usávamos os cofres; quando queríamos que uma carta chegasse ao destino sem ser aberta, usávamos lacre; quando queríamos garantir que um destinatário recebia uma carta, enviávamo-la com aviso de recepção.

Hoje em dia, no mundo em que vivemos, será possível ter segurança? Podemos ter um cofre na Internet para guardar dinheiro virtual? Podemos assinar documentos virtuais sem que ninguém falsifique a nossa assinatura? Podemos enviar *e-mails* lacrados? Podemos enviar *e-mails* com aviso de recepção?

Estas e outras questões são o tema deste artigo.

1 Introdução

Todos nós já ouvimos falar de criptografia e codificação de mensagens. Muitos de nós até já usámos aquilo a que chamaremos sistemas criptográficos nos nossos jogos de crianças. Por isso este artigo não será uma explicação do que é a criptografia mas sim uma formalização de alguns conceitos relacionados com este tema.

Começaremos então por definir o que é um *sistema criptográfico* e em seguida falaremos de criptografia de chave privada dando alguns exemplos. Na secção 3 trataremos da cripto-análise de algumas dessas cifras, i.e., dada uma mensagem codificada, tentar descobrir qual é a chave que está a ser utilizada.

A secção 4 é dedicada à criptografia de chave pública e exemplos da sua utilização. Por fim, a última secção é uma tentativa de criação de um protocolo para a implementação de dinheiro virtual para fazer compras *on-line*. Vamos então começar por formalizar alguns conceitos e notação que vamos utilizar ao longo deste artigo.

1.1 Conceitos e Notação

Como o objectivo principal da criptografia é permitir que duas pessoas comuniquem por um canal inseguro, por exemplo uma linha telefónica ou uma rede de computadores, de tal forma que um intruso não consiga perceber o que está a ser dito, vamos precisar de dois comunicadores e de um intruso. Vamos chamar aos comunicadores *A*na e *B*runo e ao intruso, que vai tentar interceptar a comunicação, *C*arlos.

À mensagem original chamaremos *mensagem* e representaremos por letras minúsculas e à mensagem que a Ana envia de facto chamaremos *mensagem codificada* e será representada por letras maiúsculas.

Vamos então começar por ver como é que os comunicadores podem codificar e descodificar as mensagens.

Definição 1. Um *sistema criptográfico* é um quíntuplo $\langle X, Y, K, E, D\rangle$ em que

- X é o conjunto das *mensagens*, finito;
- Y é o conjunto das *mensagens codificadas*, finito;
- K é o conjunto de todas as chaves possíveis, finito;
- $E = \{e_k : X \to Y\}_{k\in K}$ é o conjunto das funções de codificação;
- $D = \{d_k : Y \to X\}_{k\in K}$ é o conjunto das funções de descodificação;

tal que qualquer que seja $k \in K$

$$\forall x \in X \quad d_k(e_k(x)) = x. \tag{1}$$

A Ana, para codificar uma mensagem x com a chave k_1, faz $e_{k_1}(x)$. O Bruno, para descodificar uma mensagem y com a chave k_2, faz $d_{k_2}(y)$.

Vemos facilmente que e_k tem de ser injectiva, o que sai directamente de (1). Se assim não fosse a descodificação poderia ser ambígua. Suponhamos que existiam duas mensagens x e x' tais que $e_k(x) = e_k(x') = y$. Como deveria o Bruno descodificar y? Como x ou como x'?

Também temos que decidir o que é o conjunto X. Podemos pensar em X, conjunto das mensagens, como sendo todas as palavras do dicionário, ou podemos pensar em X como sendo o conjunto das letras do alfabeto. Neste último caso teríamos de definir o que seria $e_k(x_1x_2\ldots x_n)$, ou seja, como é que codificaríamos as palavras. Podemos ainda considerar X um conjunto de números e a cada letra associar um número, $\mathtt{a} \mapsto 0$, $\mathtt{b} \mapsto 1$, $\ldots$, $\mathtt{z} \mapsto 25$. Mais à frente veremos exemplos em que usamos esta técnica.

Um dos objectivos é que os nossos sistemas criptográficos sejam práticos e fáceis de utilizar. Para isso é necessário que as funções e_k e d_k sejam facilmente computáveis.

Outro objectivo é que o sistema criptográfico seja seguro: ao ver uma mensagem codificada y, o Carlos não deve conseguir descobrir $k \in K$ e consequentemente a mensagem original x. Vamos então ver alguns exemplos de sistemas criptográficos.

2 Criptografia de Chave Privada

A criptografia de chave privada é uma forma de codificação em que os dois comunicadores, a Ana e o Bruno, escolhem uma chave $k \in K$ quando estão juntos, ou quando têm um canal seguro à disposição para as suas comunicações. O protocolo de comunicação é muito simples: se a Ana quiser enviar uma mensagem x ao Bruno usa a função e_k para codificar a mensagem, $e_k(x) = y$, e o Bruno de seguida usa a função d_k para descodificar a mensagem codificada, $d_k(y) = d_k(e_k(x)) = x$.

$$\boxed{A \xrightarrow{y=e_k(x)} B}$$

Como o Carlos não conhece a chave, não consegue descodificar as mensagens. Na próxima secção veremos como é que o Carlos pode tentar descobrir as chaves utilizadas pela Ana e pelo Bruno, técnica que é denominada cripto-análise.

Vamos então ver algum exemplos de cifras. Começamos pela mais antiga: a *cifra de translação*.[1]

Definição 2 (Cifra de Translação). Seja $X = Y = K = \mathbb{Z}_m$. Para cada $k \in K$, $0 \leq k \leq m$ define-se[2]

$$e_k(x) = x + k \bmod m;$$
$$d_k(y) = y - k \bmod m.$$

[1] Shift cipher.

[2] Quando usamos $k = 3$ esta cifra tem o nome de *cifra de César* precisamente por ter sido usada pelo imperador romano na codificação das suas mensagens.

Podemos usar o número m que quisermos dependendo do número de caracteres que queremos codificar. Normalmente esta cifra é usada com $m = 26$ pois são os caracteres do alfabeto. Não é costume codificar os espaços e outros caracteres, pois se os usarmos facilitamos a cripto-análise.

Vamos usar a convenção de a cada letra associar um número de acordo com a Tabela 1.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Tabela 1: Conversão entre Números e Letras

Esta cifra é muito simples de utilizar. Se quisermos codificar uma mensagem $x_1x_2 \dots x_n$ fazemos $e_k(x_1x_2 \dots x_n) = e_k(x_1)e_k(x_2) \dots e_k(x_n)$.

Exemplo 1. Usando $k = 4$ a mensagem `ola` será codificada como

$$\begin{array}{lll} \texttt{o} \mapsto 14 & e_4(\texttt{o}) \mapsto e_4(14) = 14 + 4 \equiv 18 \bmod 26 & 18 \mapsto \texttt{S} \\ \texttt{l} \mapsto 11 & e_4(\texttt{l}) \mapsto e_4(11) = 11 + 4 \equiv 15 \bmod 26 & 15 \mapsto \texttt{P} \\ \texttt{a} \mapsto \ 0 & e_4(\texttt{a}) \mapsto e_4(0) \ = \ 0 + 4 \equiv \ 4 \bmod 26 & 4 \mapsto \texttt{E} \end{array}$$

dando origem à mensagem codificada `SPE`.

A *cifra afim*[3] é uma generalização da cifra de translação:

Definição 3 (Cifra Afim). Seja $X = Y = \mathbb{Z}_m$ e $K = \{(a, b) \in \mathbb{Z}_m \times \mathbb{Z}_m : \gcd(a, m) = 1\}$.[4] Para cada $(a, b) \in K$ define-se

$$\begin{aligned} e_k(x) &= ax + b \bmod m; \\ d_k(y) &= a^{-1}(y - b) \bmod m. \end{aligned}$$

Exemplo 2. Vamos codificar de novo a mensagem do exemplo anterior usando $k = (5, 3)$ como chave,

$$\begin{array}{lll} \texttt{o} \mapsto 14 & e_{(5,3)}(\texttt{o}) \mapsto e_{(5,3)}(14) = 5 \times 14 + 3 \equiv 21 \bmod 26 & 21 \mapsto \texttt{V} \\ \texttt{l} \mapsto 11 & e_{(5,3)}(\texttt{l}) \mapsto e_{(5,3)}(11) = 5 \times 11 + 3 \equiv \ 6 \bmod 26 & 6 \mapsto \texttt{G} \\ \texttt{a} \mapsto \ 0 & e_{(5,3)}(\texttt{a}) \mapsto e_{(5,3)}(0) \ = 5 \times \ 0 + 3 \equiv \ 3 \bmod 26 & 3 \mapsto \texttt{D} \end{array}$$

obtendo-se assim a mensagem codificada `VGD`.

[3] Affine cipher.

[4] K não é definido como sendo $\mathbb{Z}_m \times \mathbb{Z}_m$ pois é necessário existir a^{-1} para que a equação esteja bem definida e isto é verificado precisamente quando $\gcd(a, m) = 1$.

A cifra de translação é um caso particular desta fazendo $a = 1$ e $b = k$. Uma das cifras mais usadas em puzzles cripto-numéricos de jornais e revistas é a cifra de que vamos falar agora, a *cifra de substituição*.[5]

Definição 4 (Cifra de Substituição). Seja $X = Y = \mathbb{Z}_m$ e $K = \{\pi : \pi \text{ é uma permutação sobre } \mathbb{Z}_m\}$. Para cada permutação $\pi \in K$ define-se

$$e_k(x) = \pi(x);$$
$$d_k(y) = \pi^{-1}(y).$$

Exemplo 3. Vamos codificar de novo a mensagem `ola` usando a seguinte permutação π:

a	b	c	d	e	f	g	h	i	j	k	l	m
Y	G	N	S	U	H	Q	A	O	I	W	C	J

n	o	p	q	r	s	t	u	v	w	x	y	z
T	M	R	X	D	L	Z	E	V	F	K	B	P

$$e_\pi(\texttt{o}) = \texttt{M}, \quad e_\pi(\texttt{l}) = \texttt{C}, \quad e_\pi(\texttt{a}) = \texttt{Y}$$

i.e., a codificação da mensagem `ola` dá origem à mensagem codificada `MCY`.

A cripto-análise desta cifra parece ser bastante difícil devido ao número de chaves possíveis, no entanto veremos na próxima secção que, recorrendo a métodos estatísticos, é bastante fácil.

Os sistemas que usámos até agora codificam sempre um dado símbolo da mesma maneira (quando usamos a mesma chave). Estes sistemas são chamados *sistemas criptográficos mono-alfabéticos*.

Vamos referir ainda a título de curiosidade duas outras cifras bastante conhecidas que não são mono-alfabéticas, a *cifra Vigenère* e a *cifra matricial* ou *cifra de Hill*.[6]

Definição 5 (Cifra Vigenère). Seja $X = Y = K = (\mathbb{Z}_m)^n$, para um n fixo. Para cada $k = (k_1, k_2, \ldots, k_n) \in K$ define-se

$$e_k(x_1, x_2, \ldots, x_n) = (x_1 + k_1 \bmod m, \ldots, x_n + k_n \bmod m);$$
$$d_k(y_1, y_2, \ldots, y_n) = (y_1 - k_1 \bmod m, \ldots, y_n - k_n \bmod m).$$

Definição 6 (Cifra Matricial). Seja $X = Y = (\mathbb{Z}_m)^n$ para um n fixo, $K = \{\text{matrizes invertíveis } n \times n \text{ com componentes em } \mathbb{Z}_m\}$. Para cada $k \in K$ define-se

$$e_k(x_1, x_2, \ldots, x_n) = (x_1, x_2, \ldots, x_n) \cdot k;$$
$$d_k(y_1, y_2, \ldots, y_n) = (y_1, y_2, \ldots, y_n) \cdot k^{-1}.$$

[5] Substitution cipher.
[6] Lester S. Hill.

Estas duas cifras codificam blocos de comprimento n. Por codificarem blocos em vez de caracteres isolados, é possível que um dado carácter tenha duas codificações diferentes. A estes sistemas que podem codificar um símbolo de mais do que uma maneira usando a mesma chave chamam-se *sistemas criptográficos poli-alfabéticos*.

Exemplo 4. Se codificarmos a palavra `ele` $\mapsto (4, 11, 4)$ usando a cifra Vigenère com a chave $k = (1, 2, 3)$ obtemos como resultado $(4, 11, 4)+(1, 2, 3) \equiv (5, 13, 7) \bmod 26$, que corresponde à mensagem codificada `FNH`, ou seja o primeiro `e` foi transformado em `F` e o segundo em `H`.

3 Cripto-Análise

Quando se estuda um sistema criptográfico em termos da sua segurança é importante definirmos à partida aquilo que um intruso conhece do sistema. Normalmente estes estudos têm subjacente o chamado *princípio de Kerckoff*, que diz que qualquer indivíduo que queira interceptar a comunicação entre outros dois pode ter toda a informação possível acerca do sistema criptográfico utilizado por eles.

Poderíamos não assumir isto e então grande parte da segurança da comunicação estaria associada ao desconhecimento sobre o sistema envolvido. Não queremos que a tarefa do intruso seja facilitada pelo simples facto de descobrir o sistema criptográfico em utilização. Sendo assim, a tarefa do cripto-analista é a seguinte: dada uma mensagem codificada tentar descobrir qual a chave de codificação usada pelos dois comunicadores sabendo o sistema criptográfico que estão a usar. Vamos então referir quatro tipos possíveis de ataque a um sistema criptográfico.

1. **Conhecimento de uma mensagem codificada:**[7] o intruso tem conhecimento de um exemplo de uma mensagem codificada, i.e., sabe $d_k(x)$ para algum $x \in X$.

2. **Conhecimento de uma mensagem:**[8] o intruso tem conhecimento de uma mensagem $x \in X$ e da sua codificação $e_k(x)$.

3. **Possível escolha de mensagens:**[9] o intruso tem acesso temporário à máquina de codificação e consegue codificar as mensagens $x \in X$ que quiser obtendo a sua codificação $e_k(x)$.

[7] Ciphertext-only attack.
[8] Known plaintext attack.
[9] Chosen plaintext attack.

4. **Possível escolha de mensagens codificadas:**[10] igual ao anterior mas usando a máquina de descodificação, i.e., descobrindo os pares $(y, d_k(y))$.

Independentemente do ataque que faremos, temos de decidir qual a estratégia a seguir para cripto-analisar uma mensagem codificada. Um método possível para a cripto-análise é o chamado *método da força bruta* que consiste em testar todas as chaves possíveis para ver qual é que está a ser utilizada. No entanto, quando o número de chaves possíveis é muito elevado seguimos por vezes outra estratégia que é o *método da análise de frequências* das letras na língua. Apesar deste método ter grandes vantagens face ao anterior, a sua aplicação não é sempre possível.

3.1 Método da Força Bruta

Vamos então fazer a cripto-análise de um dos exemplos da secção anterior usando o método da força bruta. Escolhamos o Exemplo 1.

Ao olhar para a definição de cifra de translação verificamos que o seu número de chaves é reduzido, tem apenas m chaves. No nosso caso como queremos codificar as letras do alfabeto temos apenas 26 chaves. Este é um caso em que a técnica da *força bruta* é uma boa maneira de tentar descobrir a chave utilizada. Esta cifra é muito frágil mesmo com o ataque mais simples como se pode ver no exemplo que se segue.

Relembremos que a mensagem `ola` foi codificada na mensagem `SPE`. Vamos então fazer $d_k(\texttt{SPE})$ com $k = 0,\ 1, \ldots,\ 25$.

$k = 0$	`spe`	$k = 7$	`lix`	$k = 14$	`ebq`	$k = 20$	`yvk`
$k = 1$	`rod`	$k = 8$	`khw`	$k = 15$	`dap`	$k = 21$	`xuj`
$k = 2$	`qnc`	$k = 9$	`jgv`	$k = 16$	`czo`	$k = 22$	`wti`
$k = 3$	`pmb`	$k = 10$	`ifu`	$k = 17$	`byn`	$k = 23$	`vsh`
$k = 4$	`ola`	$k = 11$	`het`	$k = 18$	`axm`	$k = 24$	`urg`
$k = 5$	`nkz`	$k = 12$	`gds`	$k = 19$	`zwl`	$k = 25$	`tqf`
$k = 6$	`mjy`	$k = 13$	`fcr`				

Verificamos que o único valor que faz sentido é $k = 4$, logo descodificaremos `SPE` para `ola` pois é a única solução aceitável. Esta técnica é eficiente, mas quando o número de chaves aumenta torna-se impraticável.

[10] Chosen ciphertext attack.

3.2 Método da Análise de Frequências

Este método consiste em estabelecer uma tabela da frequência das letras na língua portuguesa, através da leitura de jornais, revistas, etc., contando-se quantas vezes cada letra aparece nas respectivas notícias. Esta contagem não é feita recorrendo apenas ao dicionário pois este apresenta todas as palavras da língua portuguesa, mas não indica a frequência com que estas são usadas. Além da análise da frequência de cada uma das letras, devemos também analisar a frequência dos digramas e dos trigramas na língua, sequências de duas e três letras respectivamente.

Nota 5. Para aplicar esta técnica deve ser criada uma tabela recorrendo a textos da matéria em questão. Refere-se como exemplo a língua portuguesa mas se a mensagem fosse sobre aviões deveríamos recorrer a uma tabela de frequências criada a partir da análise de textos sobre aviões e se por acaso fosse sobre um tema científico específico deveríamos usar conhecimento sobre textos desse domínio. De referir ainda que a primeira análise das letras de um texto foi feita pelos árabes que contaram a frequência das letras no Corão.

Letra	**Freq. Relativa**	**Letra**	**Freq. Relativa**
a	0.1356	l	0.02760
e	0.1241	v	0.01460
o	0.1092	g	0.01215
s	0.0779	q	0.01009
i	0.0686	f	0.00980
r	0.0678	b	0.00934
n	0.0557	h	0.00724
d	0.0528	z	0.00427
t	0.0522	j	0.00365
c	0.0436	x	0.00225
m	0.0418	k	0.00052
u	0.0404	y	0.00036
p	0.0280	w	0.00029

Tabela 2: Frequências das Letras na Língua Portuguesa

A Tabela 2 foi obtida através da contagem das letras das palavras saídas em todos os jornais com publicações *on-line* desde 1991.[11] Apresenta-se ainda a mesma contagem para os digramas e trigramas mais comuns na língua portuguesa, na Tabela 3.[12]

[11] Dados fornecidos pelo Laboratório de Sistemas de Linguagem Falada do INESC.

[12] As tabelas apresentadas foram construídas recorrendo à análise de aproximadamente

Digrama	**Freq. Relativa**	**Trigrama**	**Freq. Relativa**
de	0.0249	ent	0.0142
es	0.0224	que	0.0118
os	0.0199	nte	0.0093
nt	0.0196	res	0.0067
ra	0.0191	est	0.0065
en	0.0185	nto	0.0065
do	0.0181	com	0.0063
co	0.0173	con	0.0063
te	0.0166	ado	0.0062
ar	0.0165	sta	0.0059
re	0.0165	ara	0.0058
as	0.0164	par	0.0057

Tabela 3: Digramas e Trigramas Mais Comuns na Língua Portuguesa

O objectivo desta técnica é analisar a frequência de uma dada letra na mensagem codificada e tentar inferir através das tabelas acima qual será a sua descodificação. Assim começamos por procurar a letra mais frequente na mensagem codificada e é natural supor que essa letra seja a codificação do a pois esta é a letra mais frequente na língua portuguesa. Devemos ter sempre em conta que esta é a hipótese mais provável, no entanto podem existir mensagens em que a frequência das letras não corresponde à frequência apresentada na tabela.

Repetimos o processo até determinar a descodificação de todas as letras da mensagem codificada. Podemos ainda usar a tabela dos digramas e dos trigramas para inferir sobre a mensagem original.

Esta técnica é particularmente adequada para fazer a cripto-análise das mensagens codificadas com a cifra de substituição pois esta cifra tem $m!$ chaves o que torna a técnica anterior impraticável.

Também é possível fazer a cripto-análise da cifra afim usando esta técnica. Para isso escolhemos as duas letras mais comuns na mensagem codificada, l_1 e l_2, (chamemos y_1 e y_2 aos seus códigos de acordo com a Tabela 1) e as duas letras mais comuns na língua portuguesa, a e e, como se vê na Tabela 2 (cujos códigos são respectivamente $x_1 = 0$ e $x_2 = 4$). Seguidamente tentamos resolver o sistema

$$\begin{cases} y_1 = ax_1 + b \bmod 26 \\ y_2 = ax_2 + b \bmod 26 \end{cases}.$$

100 000 palavras diferentes. Podemos referir a título de curiosidade que a palavra de foi a palavra que apareceu mais vezes, 18 859 628 vezes.

Tentamos assim ver se l_1 é a codificação da letra `a` e l_2 é a codificação de `e`.

1. Se $\gcd(a, 26) = 1$, a e b são possíveis candidatos a chave e então descodificamos toda a mensagem com esta chave.

   (a) Se o resultado da descodificação for satisfatório aceitamos esta chave.

   (b) Se o resultado não for satisfatório, descodificámos para uma mensagem sem sentido, vamos procurar outra chave escolhendo por exemplo a primeira e terceira letra mais comuns no alfabeto `a` e `o` e tentamos resolver de novo o sistema usando $x_1 = 0$ e $x_2 = 14$, respectivamente os códigos de `a` e `o`. Repetimos o processo até encontrarmos a chave certa, escolhendo letras diferentes mas de modo a que as frequências destas na mensagem e na Tabela 2 sejam semelhantes.

2. Se $\gcd(a, 26) \neq 1$ escolhemos um novo par de letras como indicado acima.[13]

A cripto-análise da cifra matricial parece bastante complicada. No entanto, se em vez de usarmos o ataque **conhecimento de uma mensagem codificada**, como até agora, usarmos um ataque do tipo **conhecimento de uma mensagem**, verificamos que a única dificuldade para calcular a chave é fazer a inversão de uma matriz.

Já vimos como fazer a cripto-análise de algumas cifras. Será possível fazer a cripto-análise para qualquer mensagem? Nos anos 50 Shannon provou que para conseguir segurança incondicional é necessário que o comprimento da chave seja igual ao comprimento da mensagem. Este facto não seria um grande problema, pois poderíamos combinar uma chave muito grande e usá-la em qualquer comunicação. No entanto outro factor necessário à segurança é que cada chave seja usada uma só vez. Logo a segurança incondicional tornou-se impossível de obter. Ou seja, qualquer mensagem que seja transmitida usando um sistema criptográfico de chave privada, usando uma chave que não satisfaça os requisitos apresentados, pode ser cripto-analisada.

4 Criptografia de Chave Pública

O paradigma da *criptografia de chave pública* apareceu em 1976, criado por Diffie e Hellman. Esta nova forma de codificação das mensagens consistia

[13] Se $\gcd(a, 26) \neq 1$ é óbvio que não temos uma chave pois, por definição de K, as chaves são os pares da forma $(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26}$ tais que $\gcd(a, 26) = 1$.

em codificar as mensagens usando funções de um tipo especial, as *funções de sentido único*.[14] Este conceito é definido informalmente como se segue:

DEFINIÇÃO 7. Uma função $f : X \to Y$ diz-se uma *função de sentido único*, ou simplesmente *fsu*, se é fácil computar $f(x)$ para qualquer $x \in X$ e é difícil calcular $f^{-1}(y)$ para a maioria dos elementos $y \in Y$ escolhidos aleatoriamente.

Observação. Nada se sabe sobre a existência de *fsu* pois este facto está intrinsecamente relacionado com um dos problemas fundamentais da ciência da computação, o problema de saber se $P = NP$.[15] Sabe-se que se $P = NP$ não existem *fsu*; mas se $P \neq NP$ nada se sabe. Quem quiser aprofundar esta questão pode consultar [Lub96].

Apesar de esta ideia apenas ter sido explorada a partir de 1976, a primeira referência ao uso de *fsu* aparece num artigo de 1968. Nesse artigo a criptografia era aplicada ao armazenamento de passwords em computadores. A ideia era muito simples: ao introduzir um novo utilizador, a password deste em vez de ser gravada como foi introduzida, era codificada usando uma certa função de codificação e esse valor ficava associado ao *login* do utilizador na lista de passwords que se encontra no disco do computador. O processo de autorização tornava-se assim bastante simples. Para aceder ao computador, um utilizador introduzia o seu *login* e a sua password, aplicava-se de novo a mesma função de codificação e verificava-se se o resultado coincidia com o que estava nos registos.

O processo de violação das passwords parece bastante fácil pois qualquer intruso teria acesso à lista das passwords codificadas e ainda ao algoritmo de codificação. No entanto as funções usadas para codificar eram do tipo referido acima, logo muito difíceis de inverter. Pela primeira vez aparecia o conceito de *fsu* aplicado a uma área particular da criptografia.

O primeiro algoritmo deste tipo, perfeitamente detalhado, é da autoria de Prudy, em 1974. Neste caso as passwords eram inteiros $\mathrm{mod} 2^{64} - 59$ e a função usada era

$$f(x) = x^{2^{24}+17} + a_1 x^{2^{24}+3} + a_2 x^3 + a_3 x^2 + a_4 x + a_5,$$

onde $a_1, a_2, \ldots, a_5$ são inteiros arbitrários com 19 dígitos.

As funções de codificação que iremos usar são uma subclasse das *fsu* pois tem de ser possível ao "legítimo" receptor descodificá-las, i.e., a inversão

[14] One-way function.

[15] P é a classe dos problemas que são resolúveis em tempo polinomial por uma máquina de Turing. NP é a classe dos problemas que podem ser resolvidos em tempo polinomial por uma máquina de Turing não determinista.

destas tem de ser impossível para todos à excepção do destinatário da mensagem. Assim as funções de codificação são *fsu* desde que nos falte alguma informação, a chave de descodificação. Sabendo esta, a inversão torna-se bastante simples.

Como é óbvio estas funções de codificação não são escolhidas ao acaso. Normalmente a sua inversão tem subjacente um problema que se sabe ser de difícil resolução, usualmente NP-completo. O primeiro e mais famoso algoritmo de chave pública é o RSA, baseado na factorização prima de um número grande, com aproximadamente 200 dígitos (cf. [Rei01]). Adiante veremos um outro algoritmo, o El Gamal, que é baseado no problema do logaritmo discreto, também este de difícil resolução.

Apesar de ter sido inventado apenas nos anos 70, o RSA é um algoritmo que usa matemática anterior ao século XX. Porque é que a ideia da chave pública e o algoritmo RSA não apareceram mais cedo?

Duas respostas podem ser dadas para esta pergunta. Uma primeira é que até à década de 70 a criptografia apenas era usada pelos serviços diplomáticos e militares e talvez por isso o sistema de chave privada fosse eficaz, pois não havia a necessidade de todos comunicarem com todos. Outra resposta possível é que a eficácia do algoritmo RSA depende da utilização de números primos grandes. Sem os computadores o cálculo da potenciação de grandes números é difícil.

Uma das grandes vantagens da *criptografia de chave pública* face à *criptografia de chave privada* é que quaisquer duas pessoas podem enviar mensagens confidenciais sem terem combinado uma chave de codificação. Mais ainda, uma pessoa pode enviar mensagens codificadas a outra sem nunca a ter contactado.

Como se processa então a codificação usando um protocolo de chave pública? Basta que o receptor da mensagem publique o seu algoritmo de codificação (parte pública) e todas as pessoas que queiram comunicar com ele utilizem esse algoritmo para codificar as mensagens. A única pessoa que conhece o algoritmo de descodificação (parte privada) é o receptor, e logo apenas ele pode descodificar as mensagens enviadas usando aquele algoritmo de codificação.

Depois desta breve introdução podemos pensar como é que a chave pública nos pode ajudar a resolver alguns problemas que nos são colocados hoje em dia, nomeadamente:

- Transmissão confidencial de mensagens.
- Autenticação: poder verificar que uma mensagem foi de facto enviada por quem se identifica como emissor.

- Não-repúdio: uma pessoa não poder negar que enviou de facto uma mensagem.
- Troca de chaves: duas pessoas combinarem publicamente uma chave para usar num sistema de chave privada.
- Partilha de segredos: um sistema só poder funcionar com k autorizações e se apenas tivermos $k-1$ não conseguimos fazer nada.

Abordaremos seguidamente os quatro primeiros pontos, dando exemplos para cada um.

4.1 Transmissão Confidencial de Mensagens

Como exemplo vamos ver o algoritmo de codificação El Gamal. Este algoritmo é baseado na dificuldade de resolução do problema do logaritmo discreto, PLD. Sem entrar em grandes detalhes sobre as condições do problema, assumindo que são de fácil verificação, o PLD pode ser posto como se segue.

Definição 8 (Problema do Logaritmo Discreto). Seja p primo, $\alpha \in \mathbb{Z}_p$ um elemento primitivo[16] e $\beta \in \mathbb{Z}_p^*$.[17] O objectivo é encontrar o único inteiro a com $0 \leq a \leq p-2$ e tal que

$$\alpha^a \equiv \beta \bmod p.$$

Dizemos neste caso que $a = \log_\alpha \beta$.

Este problema é considerado de difícil resolução porque não existe nenhum algoritmo polinomial que o resolva para valores de p elevados (números com aproximadamente 150 dígitos) e tais que $p-1$ tenha pelo menos um factor primo grande.

Como é que nós podemos dizer que este problema nos dará uma boa função de codificação? Tal como é posto, o problema de calcular o logaritmo discreto de um número é complicado, mas calcular a potência, operação inversa, é muito simples. Logo podemos usar como função de codificação a potência, fácil de calcular, e como função de descodificação o logaritmo. Dito de outra forma, o logaritmo é uma *fsu*.

Definição 9 (Cifra El Gamal). Seja p primo tal que o PLD é intratável em $\mathbb{Z}_p$ e $\alpha \in \mathbb{Z}_p^*$ um elemento primitivo. Seja ainda $X = \mathbb{Z}_p^*$, $Y = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ e $K = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \bmod p\}$. Os parâmetros públicos são p, α e β. O

[16] α diz-se um elemento primitivo de $\mathbb{Z}_p$ se $\{\alpha^i : 0 \leq i \leq p-2\} = \mathbb{Z}_p^*$.
[17] $\mathbb{Z}_p^* = \{x \in \mathbb{Z}_p : x \neq 0\}$.

número a é privado. Seja ainda $z \in \mathbb{Z}_p$ um número aleatório escolhido pelo emissor. Definimos então

$$e_k(x, z) = (y_1, y_2) \text{ em que } \begin{cases} y_1 = \alpha^z \bmod p \\ y_2 = x\beta^z \bmod p \end{cases} ; \tag{2}$$

$$d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p. \tag{3}$$

Vamos ver que quem souber a consegue descodificar a mensagem.

$$y_2(y_1^a)^{-1} \equiv x\beta^z(\alpha^{az})^{-1} \qquad \text{por (2)} \tag{4a}$$

$$\equiv x\beta^z(\beta^z)^{-1} \qquad \text{por definição de } K \tag{4b}$$

$$\equiv x \bmod p \tag{4c}$$

O que o Bruno faz ao usar este algoritmo é escolher um $z \in \mathbb{Z}_p$ aleatório e "esconder" a mensagem x usando β^z. Depois envia este valor, $x\beta^z$, juntamente com α^z o que permite à Ana calcular β^z, (4b), parte fundamental do algoritmo. A Ana consegue assim descodificar a mensagem pois conhece a, enquanto o Carlos não. Assim o Carlos tem de calcular z tentando resolver o PLD pois sabe o valor de α e $y_1 = \alpha^z$.

Como vemos, se alguém souber o valor de z consegue descodificar a mensagem pois pode começar o processo a partir de (4b); por isso o valor de z também tem de ser guardado pelo emissor. Um factor que pode dificultar ainda mais a tentativa de descodificação "ilegítima" desta cifra é o facto de ser não determinística. A mensagem `ola` não é codificada sempre da mesma maneira, pois depende do z que o emissor escolher.

4.2 Assinaturas Digitais e Problemas de Autenticação

Um problema que existe na comunicação digital é que devido ao facto de os comunicadores não se verem, as mensagens que estes trocam podem estar a ser enviadas ou alteradas por alguém. Consideremos então estes dois problemas:

A. Problema da Assinatura: a Ana pretende ter a certeza que a mensagem enviada pelo Bruno não foi alterada pelo Carlos.

B. Problema da Autenticação: a Ana quer ter a certeza que foi de facto o Bruno que lhe enviou uma dada mensagem e não o Carlos fingindo ser o Bruno.

Temos assim que discutir uma nova tarefa da criptografia: garantir que o nosso emissor ou receptor é de facto quem pensamos. Para isso temos de

definir, mais uma vez informalmente, um novo conceito que é o conceito de *função de dispersão*.[18] Estas funções são normalmente de domínio público.

DEFINIÇÃO 10. Diz-se que $H : X^k \to X^l$ com $k > l$ é uma *função de dispersão* se tiver as seguintes propriedades:

1. É fácil calcular $H(x)$ para qualquer $x \in X^k$;
2. É difícil encontrar dois valores $x, x' \in X^k$ tais que $H(x) = H(x')$;
3. Dado $y \in \mathrm{Im}(H)$ não é fácil determinar $x \in X^k$ tal que $H(x) = y$.

Estas funções são utilizadas para produzir assinaturas digitais pois *comprimem* as mensagens, ou seja, recebem mensagens de k símbolos e transformam-nas em mensagens de l símbolos, $k > l$. A mensagem assim obtida pode ser usada como assinatura da mensagem original, pois é mais pequena que esta.

A segunda condição da definição acima faz com que, apesar de a função não ser obrigatoriamente injectiva, seja complicado encontrar duas mensagens que tenham a mesma assinatura. A terceira condição faz com que seja difícil arranjar uma mensagem sabendo qual é a sua assinatura.

Suponhamos então que o Bruno pretende enviar uma mensagem para a Ana. Como é que ela pode ter a certeza que a mensagem recebida foi mesmo enviada por ele e não foi alterada pelo Carlos (problemas A e B identificados acima)?

4.2.1 Problema da Assinatura

O que o Bruno tem a fazer para resolver este problema é em vez de enviar apenas a mensagem, enviar a mensagem e a sua assinatura, i.e., enviar o par $(x, H(x))$. Para a Ana verificar que a mensagem não foi alterada, basta-lhe aplicar a mesma função H à primeira componente do par (a mensagem) e comparar o resultado com a segunda componente (a assinatura). Se for igual, é porque a mensagem não foi alterada. Mas como é que perante esta igualdade podemos ter a certeza que a mensagem não foi alterada?

Podemos, porque pela segunda condição da Definição 10 o Carlos não consegue encontrar outra mensagem x' tal que $H(x) = H(x')$. Ele pode apenas tentar alterar ambas as componentes do par $(x, H(x))$ gerando uma mensagem x' e calculando a respectiva assinatura. Temos por isso de resolver o problema que resulta do facto de o Carlos poder enviar uma mensagem totalmente nova, $(x', H(x'))$.

[18] Hash-function.

4.2.2 Problema da Autenticação

Para resolver este problema necessitamos apenas que o Bruno use um sistema criptográfico endomórfico.

Definição 11. Um sistema criptográfico $\langle X, Y, K, E, D\rangle$ diz-se *endomórfico* se $X = Y$.

Lema 12. *Seja $f : I \to I$ uma função injectiva e I um conjunto finito. Então f é um isomorfismo.*

Observação. Devido ao sistema criptográfico usado pelo Bruno ser endomórfico e a respectiva função de codificação e_B ser injectiva, o Lema 12 garante que e_B é um isomorfismo e que a sua inversa é d_B. Temos então $e_B(d_B(x)) = x$ e $d_B(e_B(x)) = x$.

Depois desta observação é muito simples entender como é que a Ana pode estar certa que foi o Bruno que enviou a mensagem. Basta que o Bruno, em vez de enviar o par $(x, H(x))$, envie $(x, d_B(H(x)))$. O que a Ana tem a fazer para verificar que foi o Bruno é aplicar a função e_B (pública) ao segundo elemento da mensagem que recebeu. Assim obtém o valor de $H(x)$ pois pela observação anterior $e_B(d_B(H(x))) = H(x)$. Seguidamente faz o mesmo que fazia para verificar se a assinatura era válida, ou seja aplica a função H ao primeiro elemento do par e verifica se dá o valor que tinha acabado de calcular.

O único problema que ainda não tínhamos resolvido era o "risco" do Carlos alterar as duas componentes da mensagem. Verificamos agora que lhe é impossível fazer tal coisa, pois só o Bruno conhece d_B. Assim, mesmo que o Carlos encontre um novo par $(x', H(x'))$ não conseguirá calcular $d_B(H(x')$. Deste modo a Ana estará certa que quem enviou a mensagem foi de facto o Bruno. Por outro lado, o Bruno não pode negar que enviou a mensagem pois só ele conhece a função d_B, e por isso só ele poderá usá-la para "codificar" a assinatura. Estabelecemos então uma relação de compromisso e não-repúdio.

Observação. Por simplicidade referi que o Bruno tinha de usar um sistema criptográfico endomórfico, pois era necessário existir uma função que apenas o Bruno conhecesse, neste caso d_B. No entanto poderíamos apenas exigir que existissem duas funções $f, g : X \to X$ tais que f fosse fácil de calcular e injectiva, $f \circ g = id_X$ e g fosse apenas do conhecimento do Bruno.

4.3 Troca de Chaves Privadas Através de Canais Públicos

Suponhamos que a Ana e o Bruno querem combinar uma chave para um sistema criptográfico de chave privada e que não se podem encontrar pessoalmente nem têm nenhuma maneira de contactar seguramente. Terão por

esse motivo de combinar a chave através de um canal público. Como poderão fazê-lo?

1. Ana e Bruno escolhem um primo p e um elemento primitivo $g \in \mathbb{Z}_p$ primitivo. O Carlos pode conhecer p e g, pois foram combinados através de um canal público.

2. Em seguida a Ana escolhe um valor $k_A < p$ e envia $g^{k_A} \bmod p$ para o Bruno.

3. O Bruno faz o mesmo e envia $g^{k_B} \bmod p$ para a Ana.

   Neste momento o Carlos pode saber os valores p, g, g^{k_A} e g^{k_B}, pois todos eles foram combinados através de um canal público.

4. Como em $\mathbb{Z}_p$ temos $(g^{k_A})^{k_B} \equiv (g^{k_B})^{k_A} \equiv g^{k_A k_B} \bmod p$, a Ana e o Bruno escolhem $g^{k_A k_B} \bmod p$ para chave.

   O Bruno consegue calcular esta chave pois conhece k_B e $g^{k_A} \bmod p$. A Ana também consegue calcular a chave pois conhece k_A e $g^{k_B} \bmod p$.

   No entanto será que o Carlos consegue calcular a chave? Será que, dados p, g, g^{k_A} e g^{k_B}, consegue calcular $g^{k_A k_B} \bmod p$? É fácil ver que não consegue, pois teria que saber k_A ou k_B — e descobrir estes números seria o mesmo que resolver o Problema do Logaritmo Discreto. Logo o Carlos nunca conseguirá descobrir a chave se a Ana e o Bruno escolherem um número primo p tal que o PLD seja intratável em $\mathbb{Z}_p$.

5 O Dinheiro Virtual e a Criptografia

O comércio virtual é um mundo em forte expansão nos nossos dias. No entanto, esta expansão tem um entrave: o meio de pagamento. Normalmente existem duas maneiras possíveis de fazer esse pagamento. Uma é o envio à cobrança, a outra é através de cartão de crédito. Enquanto a primeira é mais segura (só pagamos a encomenda quando a recebemos e o nosso número de cartão de crédito não anda a circular pela Internet, mesmo que codificado) a segunda é muito mais eficiente, barata e prática. Temos por isso de contrabalançar a segurança com a eficiência. De momento o comércio virtual ainda está a dar os primeiros passos, mas quando se desenvolver terá de haver um meio muito mais prático e seguro de fazer as transacções.

É com este objectivo que nasceu a ideia de dinheiro virtual. Será possível ter "dinheiro virtual"? Existirão "bancos virtuais"? Claro que não poderão existir bancos meramente virtuais, pois as pessoas necessitam de dinheiro

físico para a sua vida quotidiana; mas será possível termos uma conta virtual, i.e., uma conta onde cada um de nós tem dinheiro para gastar em compras *on-line*? Imaginemos que queremos comprar um livro numa livraria *on-line*. Será possível em vez de enviarmos o número do nosso cartão de crédito, enviarmos por exemplo, um ficheiro informático que valha dinheiro? Onde guardamos esse dinheiro virtual? Num "cofre virtual"?

Este será o problema que vamos abordar nesta secção. Vamos usar os termos "dinheiro físico" e "dinheiro virtual" caso se trate do dinheiro que estamos habituados ou deste novo tipo de dinheiro que queremos inventar. Vamos usar sempre algoritmos de codificação seguros. Para simplificar a abordagem ao problema vamos primeiro fazer algumas considerações sobre o dinheiro em geral e pensar em algumas propriedades que o dinheiro físico tem e que seja importante o dinheiro virtual também ter. Baseados nestas ideias criaremos um protocolo, que apesar de apresentar algumas limitações é um primeiro passo na criação de dinheiro virtual.

Antes de pensarmos como guardar o nosso dinheiro virtual, pensemos como é que este vai ser. A ideia imediata, e se calhar a única, é que as notas virtuais sejam ficheiros. Assim, teremos ficheiros nos nossos computadores que representam dinheiro e de preferência queremos que toda a gente os possa ver, ou seja, estes têm de estar numa parte pública dos nossos computadores.

Contudo, se estiverem numa zona pública dos nossos computadores poderão ser roubados. Então como é que os guardamos? O ideal é que o dinheiro virtual não necessite de ser guardado em cofres virtuais; este dinheiro devia ser ele próprio à prova de roubo. Como conseguir isso? Fazendo com que o nosso dinheiro tenha algumas propriedades anti-roubo.

5.1 Propriedades do Dinheiro Virtual

Pensemos então em algumas propriedades que o dinheiro virtual deverá ter.

- Todas as notas são emitidas por um banco, logo também no campo virtual temos de ter uma entidade emissora/reguladora.
- Todas as notas têm um número de série que as torna únicas, aqui também teremos essa numeração.
- As notas usuais são, ou pretende-se que sejam, difíceis de copiar, mas para já o dinheiro virtual corresponderá a ficheiros informáticos e consequentemente fácil de copiar. Assim a nossa entidade emissora terá também uma função reguladora: verificar que cada nota só pode ser usada para pagar uma operação, i.e., tem que manter registo do proprietário do ficheiro em cada instante.

Para escapar a este último ponto, de complicada resolução, e para simplificar o nosso problema, vamos apenas pensar nos *traveller cheques*.

- À partida temos uma grande vantagem: enquanto os cheques são pessoais e só podem ser usados uma vez, o mesmo não se passa com as notas.

- Os cheques, além de pessoais, têm o nome do proprietário. Assim, cada ficheiro terá também a informação do seu proprietário.

- No caso dos *traveller cheques* temos ainda que os assinar quando os recebemos, e assim garantimos que só pode usar aquele cheque quem reproduzir a nossa assinatura. Neste caso, será uma assinatura virtual que tem subjacente um algoritmo de assinatura seguro.

5.2 Protocolo para Implementação de Dinheiro Virtual

Na nossa transacção temos um comprador, a *A*na, um *B*anco e um *V*endedor. Temos ainda um outro elemento, o *L*adrão, que pretende "desviar" o dinheiro em alguma parte da comunicação.

Admitimos como certo que o Banco é uma entidade honesta, que sabe quem são os seus clientes e guarda as chaves públicas dos mesmos. Sempre que seja necessário o uso de chaves públicas, é ao Banco que elas vão ser pedidas. Vamos usar ainda e_X e d_X para representar as chaves de codificação e descodificação do elemento X. Vejamos então o protocolo.

1. Ana pede o dinheiro ao Banco, identificando-se.

2. O Banco gera um número aleatório n_1 que será o código da nota produzida. Mas, tal como foi referido atrás, esse código deve ser apenas conhecido pela Ana, como se fosse uma assinatura. Sendo assim, em vez de assinar a nota com o código n_1, o Banco assina-a com o código $e_A(n_1)$. Para manter a privacidade o Banco apaga n_1 dos seus registos e guarda o par $(nota, e_A(n_1))$.

Notas emitidas	**Notas em circulação**
$(nota, e_A(n_1))$	

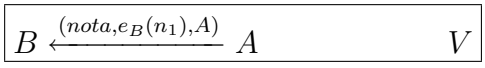
   Neste momento o Banco envia para a Ana a mensagem $(nota, e_A(n_1))$.

$B \xrightarrow{(nota, e_A(n_1))} A$	V

Isto termina a fase da emissão da nota, a qual só entrará em circulação (i.e., só será válida) quando o Banco receber a confirmação de que a Ana recebeu a nota.

3. Entramos então na fase de confirmação da recepção da nota por parte da Ana. O Banco só passa a dar a nota como válida se alguém lhe enviar o valor n_1, associado à informação da nota. Neste momento só a Ana sabe qual é o valor n_1, pois por (1) temos que $n_1 = d_A(e_A(n_1))$ e apenas a Ana conhece a função d_A. Mesmo que intercepte a nota, o Ladrão nunca conseguirá reclamá-la, pois $n_1 \neq d_L(e_A(n_1))$.

   Então o que a Ana faz é calcular o valor de n_1 e reenviá-lo codificado para o Banco, i.e., envia para o Banco a mensagem $(nota, e_B(n_1), A)$.



   É importante observar que só a Ana sabe o valor n_1, e consequentemente só ela consegue calcular $e_B(n_1)$.

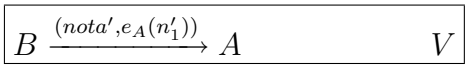
4. Ao receber esta mensagem o Banco selecciona a segunda componente da mensagem, x, e calcula $e_A(d_B(x))$. Seguidamente verifica se este é o valor associado à nota, i.e., verifica se $e_A(d_B(x)) = e_A(n_1)$.

   (a) Se for este o valor associado, ou seja, se $x = e_B(n_1)$, é porque a Ana recebeu a nota, e então ela fica válida. É adicionado ao registo das notas em circulação o triplo $(nota, e_A(n_1), A)$ e o par $(nota, e_A(n_1))$ é removido das notas emitidas. A partir deste momento o Banco só paga o valor da nota a quem provar saber o valor n_1.

Notas emitidas	**Notas em circulação**
~~$(nota, e_A(n_1))$~~	$(nota, e_A(n_1), A)$

   (b) Se não for é porque alguém se tentou passar pela Ana. Em consequência disso, a nota fica inválida e é criada uma nova nota com um novo código $(nota', e_A(n_1'))$, que é de novo enviada para a Ana.

Notas emitidas	**Notas em circulação**
~~$(nota, e_A(n_1))$~~	
$(nota', e_A(n_1'))$	



Fica assim confirmada a recepção ou o desvio da nota.

Neste momento a Ana tem o seu dinheiro e mesmo que este seja roubado ninguém o poderá usar pois não conhece o valor n_1. Vamos então passar à fase de pagamento. Esta tem que ser realizada em simultâneo pelo vendedor e pelo comprador.

5. Suponhamos então que a Ana quer comprar um livro ao Vendedor. A Ana gera um número aleatório n_2 e envia para o Vendedor o par $(nota, e_V(n_2))$.

Notas emitidas	**Notas em circulação**
	$(nota, e_A(n_1), A)$

$$B \qquad\qquad A \xrightarrow{(nota, e_V(n_2))} V$$

Segue-se de novo o protocolo de confirmação de recepção (Passos 3 e 4).

6. O Vendedor recebe o par $(nota, e_V(n_2))$ e obtém o valor n_2 calculando $d_V(e_V(n_2)) = n_2$. Em seguida reenvia para a Ana o par $(nota, e_A(n_2))$.

$$B \qquad\qquad A \xleftarrow{(nota, e_A(n_2))} V$$

7. A Ana recebe a mensagem do Vendedor e verifica se o código que vem associado à nota é n_2, calculando $d_A(e_A(n_2))$. Note-se aqui que apenas a Ana e o Vendedor conhecem n_2. Logo só o Vendedor consegue enviar $e_A(n_2)$.

   (a) Se for esse o valor associado o Vendedor recebeu a nota. Então a Ana comunica ao Banco que já não possui a nota. Para isso envia-lhe a mensagem $e_B(nota, n_1, e_V(n_2), V)$, ou seja comunica que a nota está agora em posse do Vendedor e só deverá ser paga a quem mostrar saber o valor n_2. Mostra por outro lado que a nota era dela pois também conhece n_1.

$$B \xleftarrow{e_B(nota, n_1, e_V(n_2), V)} A \qquad\qquad V$$

   O Banco altera então o registo das notas em circulação mudando o anterior registo $(nota, e_A(n_1), A)$ para $(nota, e_V(n_2), V)$.

Notas emitidas	**Notas em circulação**
	~~$(nota, e_A(n_1), A)$~~
	$(nota, e_V(n_2), V)$

(b) Se não for é porque alguém interceptou a mensagem da Ana para o Vendedor. Neste caso a Ana reenvia a mensagem e voltamos ao Ponto 5.

Nesta altura apenas a Ana e o Vendedor sabem o valor de n_2, pelas mesmas razões indicadas no Passo 3. Em todas as comunicações n_2 viajou codificado, uma vez com a chave da Ana e outras duas com a chave do Vendedor. Logo é impossível para qualquer outra pessoa descobrir o valor de n_2.

8. V envia então ao Banco a mensagem $(nota, e_B(n_2))$ e o Banco, à semelhança do que fez atrás calcula $e_V(d_B(e_B(n_2))) = e_V(n_2)$ e verifica se este é o valor associado à nota. A igualdade só se verifica se o segundo elemento do par enviado for $e_B(n_2)$.

$$\boxed{B \xleftarrow{(nota, e_B(n_2))} V}$$

(a) Se for o valor associado à nota, o Banco confirma a transacção e retira a nota de circulação.

Notas emitidas	**Notas em circulação**
	~~$(nota, e_V(n_2), V)$~~

Seguidamente, o Vendedor conclui a transacção enviando a encomenda à Ana pois o Banco já lhe creditou o dinheiro.

(b) Se não for é porque alguém está a querer passar-se por Vendedor.

Notar que é impossível a Ana reutilizar a nota pois esta sai imediatamente de circulação.

Assim damos por terminada a descrição do protocolo. Neste protocolo cada nota é usada apenas uma vez, ou seja, estas notas são semelhantes aos cheques usuais. Se quiséssemos aplicar um protocolo semelhante que permitisse a circulação de uma nota por vários donos, teríamos de alterar o Ponto 8a e em vez de tirarmos a nota de circulação, combinaríamos um novo número n_3 do conhecimento exclusivo do Banco e do Vendedor, para que a Ana não voltasse a reutilizar a nota.

6 Agradecimentos

Começo por agradecer aos Professores Amílcar Sernadas e Paulo Mateus pela troca de ideias sobre a última secção e pela ajuda na preparação do

seminário em geral. Queria ainda agradecer ao Eng. Hugo Meinedo pela disponibilização dos dados para o cálculo das tabelas de frequências apresentadas. Por fim, queria agradecer à Ana pela paciência e pela ajuda na revisão do texto deste artigo, bem como na preparação do seminário.

7 Bibliografia

Quem quiser iniciar o estudo nesta área poderá consultar [Sti95], um bom livro para começar. [Men97] é também um excelente livro sobre esta matéria, mas requer algum conhecimento prévio.

Referências

[Sti95] Douglas R. Stinson. *Cryptography — Theory and Practice*. CRC Press, 1995.

[Men97] A. Menezes, P. Van Oorschot e S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[Kob99] Neal Koblitz. *Algebraic Aspects of Cryptography*. Springer, 1999.

[Lub96] Michael Luby. *Pseudorandomness and Cryptographic Applications*. Princeton University Press, 1996.

[Rei01] Tiago Reis. Criptografia e jogos por telefone. In J. Boavida, A. Cannas da Silva, L. Cruz-Filipe, J. Fachada, and P. Resende, editors, *Seminário Diagonal — Proceedings IST 2000–01*, pages 69—76, Lisbon, Portugal, November 2001.