Técnicas Criptográficas

José Manuel E. Valença *

16 de Março de 2010

*Departamento de Informática, Universidade do Minho, Campus de Gualtar Braga

5.Criptografia de Chave Pública

5.1 Funções "one-way" e Sistemas "trapdoor"

A segurança de muitas técnicas criptográficas depende, crucialmente, da existência de certas funções matemáticas $f : X \to Y$ invertíveis que têm uma implementação computacionalmente eficiente no sentido "directo" mas cuja inversa é computacionalmente intratável.

São funções paras as quais a **computação directa**

$$(dado\ qualquer\ x \in X,\ determinar\ y \in Y\ tal\ que\ f(x) = y\ )$$

é computacionalmente simples, enquanto que a **computação inversa**

$$(dado\ qualquer\ y \in Y,\ determinar\ x \in X\ tal\ que\ f(x) = y\ )$$

é computacionalmente intratável com os recursos usuais.

Estas funções designam-se por **funções *one-way*** ou **funções unidireccionais**.

EXEMPLO 29: Embora não exista nenhuma prova formal de que existam funções unidireccionais é credível, com a experiência existente, que realmente existam tais funções.

As inversas dessas funções possuem implementações que, quase sempre, têm **complexidade sub-exponencial**. Recordemos que para descrever esse tipo de complexidade se usava a notação

$$L_n[p, c] \ = \ O(2^{c\, n^p\, (\log_2 n)^{1-p}})$$

para $p \in [0,1]$ e $c > 0$.

São candidatos a funções unidireccionais os seguintes exemplos:

Multiplicação/Factorização

A **multiplicação** de inteiros (*dados* $x, y \in \mathbb{N}$ *calcular* $z \in \mathbb{N}$ *tal que* $z = x \cdot y$) é implementável de forma eficiente mesmo para valores muito grandes de x e y. A complexidade de uma implementação comum da multiplicação é $O(n^2)$ (n o número de *bits* dos argumentos).

Quando x e y são números primos, a multiplicação tem um problema inverso: *dado* z *determinar* x *e* y *tais que* $z = x \cdot y$. Este problema chama-se **factorização** de z e os valores x e y chama-se **factores primos** de z.

Ao contrário da multiplicação, não é conhecido actualmente nenhuma implementação da factorização que possa ser considerada computacionalmente tratável com os recursos usuais. Os melhores algoritmos são sub-exponenciais: o melhor algoritmo genérico conhecido tem complexidade $L_n[1/3, c]$ com $c = (64/9)^{1/3}$ apesar de os que mais são usados terem complexidade ligeiramente superior $L_n[1/2, 1]$

Exponencial/Logaritmo Discreto

Dados um primo p grande e $0 < a < p$, a função $\exp_{p,a} : \mathbb{Z}_{p-1} \rightarrow \mathbb{Z}_p^*$ [31]

$$\exp_{p,a} : x \mapsto a^x \pmod{p}$$

é implementável de forma eficiente; de facto pode-se implementar com um algoritmo de complexidade linear ou melhor.

Prova-se também que, escolhendo uma base a apropriada, a função é um isomorfismo entre $\mathbb{Z}_{p-1}$ e $\mathbb{Z}_p^*$. Porém o problema inverso (dado y encontrar x tal que $y = a^x \mod p$) é, quanto muito, sub-exponencial. O melhor algoritmo conhecido está relacionado com o melhor algoritmo para resolver o problema da factorização e tem também complexidade $L_n[1/3, c]$ com $c = (64/9)^{1/3}$.

Raiz quadrada modular

Dado um $n = p \cdot q$, em que p, q são primos grandes, o problema é

dado um qualquer $x < n$ *determinar, se existir, um* y *tal que*

$$x = y^2 \pmod{m}$$

Prova-se que este problema é redutível ao problema da factorização de n e, por isso, a sua complexidade é idêntica à da factorização.

[31] $\mathbb{Z}_n$ designa o corpo finito definido pelos inteiros no intervalo $[0, n-1]$ equipados com as operações de soma e multiplicação. $\mathbb{Z}_p^*$ – com p primo – designa o grupo multiplicativo dos inteiros no intervalo $[1, p-1]$.

Estes exemplos apontam para funções inteiras $f\colon \mathbb{N} \longrightarrow \mathbb{N}$ mas, como iremos ver, é conveniente considerar funcionais em primeiro lugar. A formalização da noção de unidireccionalidade de uma funional $h\colon \mathbb{N} \longrightarrow \wp(\mathbb{N})$ traduz, essencialmente, a incapacidade de extrair do resultado $A = h(x)$ qualquer tipo de informação sobre o argumento x.

Outra forma de exprimir esta ideia consiste em impor, como condição de unidireccionalidade, a exigência de que os eventuais argumentos x que conduzam ao resultado $h(x)$, se "distribuam aleatoriamente".

117 DEFINIÇÃO
Dada uma funcional $h\colon \varpi \longrightarrow 2^{\varpi}$ *e* $A \in 2^{\varpi}$, *o* **rastro** *é o conjunto*

$$A^h \;=\; \{\, x \in \varpi \mid h(x) \supseteq A \,\}$$

Os "eventuais argumentos" x que produzem $A \subseteq h(x)$ são descritos pelo rastro A^h que sejam, de alguma forma, algoritmicamente aleatórios. Isto implica que é necessário especificar as computações que se consideram efectivamente computáveis e que (quer através da abordagem de Kolmogorov ou de Martin-Löf) determinam a noção de aleatoriedade algorítimica.

Outro tipo de especificação incide sobre a família U dos "resultados" u cujos rastros se pretende testar.

Usamos U para escolher o tipo de unidireccionalidade que é conveniente para cada aplicação específica do conceito.

Por exemplo, pode-se tomar uma colecção U só formada por u que sejam aleatórios. Então se h for aleatório estará garantida a unidireccionalidade em U.

Outras vezes toma-se U como um conjunto de conjuntos singulares $\{y\}$ quando y percorre um qualquer domínio enumerável. Neste caso a unidireccionalidade procura que, para qualquer y nesse domínio, o rastro $\{y\}^h$ seja aleatório.

118 Definição
A funcional h é **unidireccional** *se é efectivamente computável e, para todo A suficientemente aleatório, é aleatório o rastro A^h.*

□

Vamos agora caracterizar a unidireccionalidade das funções.

119 Definição
A função $f : \mathbb{N} \to \mathbb{N}$ é unidireccional quando é computável e, para todo $y \in \mathbb{N}$ e toda a funcional efectivamente computável F, o rastro $F(y^f)$ é pelo menos tão aleatório quanto $F(y)$.

A unidireccionalidade é uma propriedade específica dessa função e traduz, essencialmente, o facto de dado um qualquer eventual resultado y não é computacionalmente tratável encontrar um x (mesmo que existam varios) tal que $f(x) \simeq y$.

Outro tipo de problema ocorre quando se tem uma colecção destas funções $\mathcal{F} = \{f\}$ e, fixando um qualquer par de valores (x, y), procura-se encontrar um $f \in \mathcal{F}$ tal que $f(x) \simeq y$. Isto requer a noção de *sistema de chaves*

120 DEFINIÇÃO
Um ***sistema de chaves*** *é uma classe efectivamente enumerável* $K = \{k\}$ *de funções unidireccionais* $k \colon \mathbb{N} \longrightarrow \mathbb{N}$ *de tal forma que, para cada* $v \in U$, *a funcional* $k \mapsto k^{-1}(v)$ *é unidireccional.*

A funcional $k \mapsto k^{-1}(v)$ associa a cada $u \in U$ o rastro $\{\, k \in K \mid k(u) \subseteq v \,\}$. Portanto, afirmar que esta funcional é unidireccional é equivalente a exigir que, para todo $u, v \in U$, a string

$$\{\, k \in K \mid k(u) \subseteq v \,\}$$

seja aleatória[32].

Particularizando para os casos em que u, v são conjuntos singulares, com $u = \{x\}$ e $v = \{y\}$, isto significa que, para x, y arbitrários, não é possível obter informação sobre qual é a chave k que verifica $k(x) = y$.

EXEMPLO 30(FUNÇÕES HASH):
A noção de unidireccionalidade permite caracterizar formalmente as propriedades de segurança aliadas às funções de "hash". Para representar uma função de hash arbitrária $H \colon \mathbb{N} \longrightarrow \mathbb{B}^t$ usaremos duas funcionais:

[32] Esta condição exprime exprime aquilo que vulgarmente se designa por "não-invertibilidade das chaves".

1. A funcional h mapeia $x \in \mathbb{N}$ no conjunto singular $\{H(x)\}$.
2. Assume-se uma codificação sobrejectiva de pares de naturais em naturais $(x, y) \mapsto x\|y$. A funcional g define-se como

$$g(x\|y) \;=\; \begin{cases} \emptyset & \text{se} \quad H(x) = H(y) \\ \{0\} & \text{se} \quad H(x) \neq H(y) \end{cases}$$

Com estas duas funcionais as condições de segurança da função de hash H exprimem-se da seguinte forma

unidireccionalidade	$\Rightarrow$	para todo resultado $z \leq 2^t$, é aleatório o rastro $\{z\}^h$
inexistência do 2º representado	$\Rightarrow$	para todo texto $x \in \mathbb{N}$ é aleatório o rastro $h(x)^h$
inexistência de colisões	$\Rightarrow$	$\emptyset^g$ é aleatório.

Sistemas "trapdoor"

121 DEFINIÇÃO

Seja K um sistema de chaves e h uma qualquer função de hash. Um sistema de chaves T é **sistema trapdoor** *de K para h, se existe uma* **funcional geradora** $\mathcal{G}: T \to 2^K$ *que é sobrejectiva, unidireccional e, para cada $t \in T$ e $k \in \mathcal{G}(t)$, verifica $t(k(x)) \simeq h(x)$ para todo x.*

A condição "$\mathcal{G}$ é sobrejectiva" implica que a cada chave $k \in K$ corresponde pelo menos uma função t (designada por "trapdoor" de k) que verifica a relação $t \circ k = h$. O facto de se exigir que $\mathcal{G}$ seja unidireccional implica que é intratável determinar qual é esse t a partir do conhecimento de k.

Pode acontecer que cada $\mathcal{G}(t)$ é singular; isto é, existe um única chave $k \in K$ que verifica $t \circ k \simeq h$. Neste caso $\mathcal{G}$ é uma **função geradora** do sistema "trapdoor".

Porém é mais frequente que $\mathcal{G}(t)$ seja um conjunto da forma $\{ G_p(t) \mid p \in \varpi \}$ em que os G_p formam uma sequência efectivamente computável de funções unidireccionais. O tamanho $|p|$ designa-se por **parâmetro de segurança**.

Neste caso, escolher $k \in \mathcal{G}(t)$ equivale a escolher um índice p e computar $k = G_p(t)$.

□

A aplicação dos sistemas "trapdoor" reside essencialmente nos sistemas de criptografia de chave pública como veremos adiante.

Um dos desenvolvimentos mais importantes em Criptografia foi a criação de técnicas criptográficas que usam dois tipos de chaves: chaves ditas "privadas", cuja segurança assenta do facto de serem ou não conhecidas pelos agentes apropriados, e chaves ditas "privadas", cuja segurança assenta no facto de terem sido emitidas pelas autoridades apropriadas.

Essas técnicas tomam o nome genérico de **técnicas assimétricas** (devido à assimetria nas propriedades das chaves) mas são designadas também por **técnicas de chave pública**.

As técnicas assimétricas tornaram-se possíveis a partir do momento em que foram identificadas certas funções que, não só parecem ser unidireccionais[33], como um comportamento assimétrico que conduz à assimetria de chaves.

A forma como vamos formalizar as técnicas de chave pública assenta precisamente no uso de sistemas "trapdoor" $\langle \mathcal{G}, K, T, h \rangle$. Nestas técnicas o sistema trapdoor é público; portanto são públicamente conhecidas a função de "hash" h, os sistemas de chaves T e K e a funcional geradora $\mathcal{G}$.

Uma instância particular da técnica torna público um $k \in \mathcal{G}(t)$ mas mantém privado valor de t.

Antes de vermos exemplos destas técnicas, convém examinar algumas situações concretas cujo comportamento computacional é, pelo menos, semelhante ao de um sistema trapdoor teórico. Como já foi referido não está provado que existe alguma função que seja unidireccional; existe apenas famílias de funções que mostram "promessa" de vir a ser unidireccionais.

Grupos cíclicos

Um dos exemplos de estruturas algébricas que aparentemente conduzem a sistemas *trapdoor* assenta nas propriedades dos grupos cíclicos.

Como exemplo vamos considerar grupos cíclicos da forma $\mathbb{Z}_p^*$.

[33] Não existe prova de que realmente existam funções unidireccionais; existe apenas uma boa possibilidade de que certas classes de funções exibam esse comportamento.

Seja p um primo representado com, pelo menos, 512 *bits* e que verifica certas condições de segurança que analisaremos num dos capítulos seguintes.

Nestas circunstâncias, existe sempre um elemento $g \in \mathbb{Z}_p^*$ tal que, quando x percorre os inteiros $\mathbb{Z}_{p-1}$, os elementos $g^x \pmod p$ percorrem todos os inteiros $\mathbb{Z}_p^*$; tais g chamam-se *elementos primitivos* ou *geradores primitivos*.

Assim, a candidata a função de hash é, neste caso,

$$h : x \mapsto g^x \pmod p \tag{110}$$

Acredita-se que a função h seja unidireccional; isto é, é implementável mas tem uma inversa (conhecida por *logaritmo discreto*) que é supostamente intratável. Adicionalmente deve ser livre de colisões.

Os diferentes h vão ser os índices da funcional geradora.

O sistema de chaves K é indexado pelos $\beta \in \mathbb{Z}_p^*$. O sistema trapdoor T é indexado pelos $a \in \mathbb{Z}_{p-1}$ tais que $h(a)$ é também um elemento primitivo. As respectivas chaves são

$$k_\beta : x \mapsto \beta^x \pmod p \tag{111}$$

$$t_a : x \mapsto x^a \pmod p \tag{112}$$

A funcional geradora $\mathcal{G}$ é determinada pela família de funções G_h com

$$G_h\colon\ a \quad \mapsto \quad h(a^{-1}) \tag{113}$$

Facilmente se verifica que a classe $K = \{k_\beta\}$ é um sistema de chaves e, para o *hash* h, a classe $T = \{t_a\}$ é a trapdoor correspondente a k_a. De facto:

(i) Enumerando T pelo índice a, cada G_h mapeia o índice a em T no índice β da chave k_β de tal forma que $\beta = G_h(a)$ se e só se $t_a(k_\beta(x) = h(x)$ para todo x.

De facto, sempre $(\mathrm{mod}\ p)$, fazendo $\beta = h(a^{-1}) = g^{a^{-1}}$, tem-se $k_\beta(x) = g^{x\,a^{-1}}$. Donde

$$t_a(k_\beta(x)) \ = \ \left(g^{x\,a^{-1}}\right)^a \ = \ g^x \ = \ h(x)$$

(ii) A função $h : x \mapsto g^x \ (\mathrm{mod}\ p)$ é unidireccional mas, em geral, não é uma função de *hash* porque apresenta colisões. De facto, dado um qualquer x é sempre possível encontrar $x' \neq x$ que tem a mesma imagem: basta escolher um x' que verifique $x' = x \ (\mathrm{mod}\ p)$.

No entanto, se restringirmos o domínio da função a $\mathbb{Z}_{p-1}$, a função torna-se injectiva e, por isso, não pode apresentar colisões.

(iii) Cada $k_\beta : x \mapsto \beta^x \ (\mathrm{mod}\ p)$ é unidireccional; é, de facto, injectiva e a intratabilidade do problema do logaritmo discreto está na base desta afirmação.

(iv) A colecção $K = \{k_\beta\}$ "forma" um sistema de chaves porque, dado quaisquer x, y não é possível descobrir qual é o valor de β que verifica $y = \beta^x \pmod{p}$. Mais uma vez a intratabilidade do logaritmo discreto justifica esta afirmação.

(v) É intratável, dado k_β (ou, equivalentemente, dado β) determinar t_a (ou, equivalentemente, determinar a) tal que $\beta = h(a^{-1})$; tal implicaria calcular o logaritmo discreto de β.

Sistemas RSA

Se $p > q$ são dois primos grandes e sejam $n \doteq p\,q$ e $\phi = (p-1)\,(q-1)$. Determinar ϕ conhecendo n é equivalente a conhecer-se a factorização de n que, supostamente, é intratável.

Um resultado muito importante é o chamado *teorema RSA* que, para todo $a, b \in \mathbb{Z}_n$ e todo $x \in \mathbb{Z}_m$, afirma

$$a = b \pmod{\phi} \implies x^a = x^b \pmod{n}$$

Neste contexto, escolhe-se uma função de *hash* arbitrária h e define-se um sistema *trapdoor* da seguinte forma:

Funcional geradora A funcional geradora $\mathcal{G}$ é definida pela família de funções $G_\phi : \mathbb{Z}_\phi^* \to \mathbb{Z}_\phi^*$ em que

$$G_\phi(a) = a^{-1} \pmod{\phi}$$

Chaves A família $K = \{k_r\}$, com $r \in \mathbb{Z}_\phi^*$, definida por

$$k_r : x \mapsto h(x)^r \pmod{n}$$

Trapdoors A família $T = \{t_a\}$, com $a \in \mathbb{Z}_\phi^*$, definida por

$$t_a : x \mapsto x^a \pmod{n}$$

A família $K = \{k_r\}$ é (a menos de um eventual estratégia para factorizar n) um sistema de chaves porque:

(i) Todos os k_r são unidireccionais (não só devido à função de hash $h(\cdot)$ mas também a exponenciação $x \mapsto x^r$).

(ii) As funções são indistinguíveis no sentido visto no caso anterior. Isto é, dados quaisquer x, y não é computacionalmente tratável determinar o r tal que $y = h(x)^r \pmod{n}$.

Por outro lado a colecção $T = \{t_a\}$ é também um sistema de chaves porque

(i) Cada $t_a : x \mapsto x^a \pmod{n}$ é supostamente unidireccional.

(ii) Os vários t_a são indistinguíveis, no sentido em que dados quaisquer x, y não é tratável determinar a tal que $y = x^a \pmod{n}$.

Finalmente, para mostrar que T é um sistema trapdoor de K para h temos de verificar

(i) Verifica-se a relação $t \circ \mathcal{G}(t) = h$ para todo t. Cada $a \in \mathbb{Z}_\phi^*$ determina um $t_a \in T$ e o índice $r = \mathcal{G}(a) = a^{-1} \pmod{\phi}$ (ou, equivalentemente, $r\,a = 1 \pmod{\phi}$) determina o correspondente k_r. Portanto

$$t_a\left(k_r(x)\right) \;=\; \left(h(x)^r\right)^a \pmod{n} \;=\; h(x)^{r\,a} \pmod{n} \;=\; h(x)$$

(ii) É intratável determinar t_a (ou a) a partir do conhecimento de k_r (ou r) a menos que ϕ seja conhecido.

Nota muito importante

Existe uma diferença fundamental entre este dois exemplos.

No primeiro caso (problemas de logaritmo discreto) a função geradora G_h era, como exige a definição teórica, unidireccional. Por isso não compromete a segurança do sistema se for publicamente conhecida.

No segundo exemplo a função geradora G_ϕ não é unidireccional. A não invertibilidade da chave k_r para a respectiva trapdoor t_a só é possível se o valor de ϕ for mantido secreto.

Por isso, nas técnicas RSA torna-se público apenas o parâmetro de segurança $|\phi|$ mas nunca ϕ.

5.2 Técnicas Criptográficas Básicas

Determinadas técnicas criptográficas podem ser definidas directamente a partir da noção de *sistema trapdoor* abstraindo completamente em relação aos detalhes das funções usadas.

No que se segue vamos assumir uma sistema de chaves K, uma função de hash h e um sistema de trapdoors T de K para h.

Cifra assimétrica

É possível definir uma classe muito geral de cifras assimétricas (isto é, cifras que usam chaves distintas para cifrar e decifrar) de forma bastante eficiente.

Neste tipo de cifras, o criptograma de um texto $x \in \mathbb{B}^*$ é um par $y \,||\, \sigma$ em que y é a *imagem* do texto e tem o mesmo comprimento que o texto, enquanto que σ se designa-se por *redundância*.

Todas estas técnicas têm uma fase de "setup" onde são geradas as chaves privadas t e públicas k. A segurança da técnica obriga a que, a partir do conhecimento de k, não seja computacionalmente tratável determinar t.

A técnica tem, depois, os processos ("esquemas") que são executados pelos dois agentes intervenientes: o agente que "cifra" a mensagem e o agente que "decifra" a mensagem.

Protocolo 1 :

Setup

Objectivo: definir o par de chaves (k, t) , usado nos esquemas de cifrar e decifrar,

(g.1) Gerar um t aleatório, gerar um $k \in \mathcal{G}(t)$ aleatório e publicita-o.

Cifrar

Objectivo: construir o criptograma $y \| \sigma$, conhecidos o texto x e a chave k

(c.1) gerar uma string aleatória ω e calcular $\sigma \leftarrow k(\omega)$ e $\lambda \leftarrow h(\omega)$
(c.2) calcular $y \leftarrow x \oplus \lambda$, calcular e publicitar o criptograma $z \leftarrow y \| \sigma$

Decifrar

Objectivo: reconstruir o texto limpo x conhecidos a trapdoor t e o criptograma z

(d.1) recuperar as componentes individuais do criptograma $(y, \sigma) \leftarrow z$
(d.2) recuperar λ calculando $\lambda' \leftarrow t(\sigma)$.
(d.3) recuperar x calculando $x' \leftarrow y \oplus \lambda'$.

Note-se que:

(i) O passo (g.2) recupera a chave de sessão λ. De facto tem-se $\lambda = h(\omega)$, $\lambda' = t(\sigma)$ e $\sigma = k(\omega)$. Logo, usando a identidade $t \circ k \simeq h$, temos

$$\lambda' = t(\sigma) = t(k(\omega)) = h(\omega) = \lambda$$

(ii) Como $\lambda = \lambda'$ o passo (g.3) recupera x; de facto tem-se $y = x \oplus \lambda$ e $x' = y \oplus \lambda'$; logo

$$x' = y \oplus \lambda' = x \oplus \lambda \oplus \lambda' = x$$

(iii) Se a incerteza em λ for igual à incerteza em x, a cifra é equivalente à cifra de Vernam ("one-time pad") e, por isso, é perfeitamente segura.

A incerteza em λ depende directamente apenas de dois factores: a incerteza em ω e as propriedades da função de "hash" h.

A incerteza em ω depende da forma como é gerado em (c.1) mas também da unidireccionalidade de k. Se k não for unireccional, partindo do conhecimento de σ (que é público) reduz-se a incerteza em ω.

A função h pode "perder incerteza" se a incerteza no seu contradomínio for inferior à incerteza no domínio.

Assinatura de 1 *bit*

É possível definir uma classe muito geral de assinaturas digitais sobre mensagens de tamanho 1 *bit*[34] em que

o agente A tem intensão de enviar 1 bit de informação para o agente B provando a autoria da mensagem (A é o único que a pode ter enviado) e a sua integridade (o bit não é alterado).

Ao contrário dos esquemas simples de assinaturas, vamos definir um **protocolo** de 3 passos onde *B* manifesta previamente a sua disposição para receber mensagens de *A*.

Protocolo 2 :

Geração: um *TA* procede como no protocolo 1, distribui t a *A* e torna público k

Inicialização *B* gera duas strings aleatórias ω_0 e ω_1; em seguida calcula e torna públicos $\sigma_0 \leftarrow k(\omega_0)$ e $\sigma_1 \leftarrow k(\omega_1)$.

Assinatura *A* calcula $s \leftarrow t(\sigma_b)$ e torna público o par $\langle b, s \rangle$.

Verificação *B* aceita a mensagem b sse for válido $(s = h(\omega_b))$.

[34] Podem não parecer muito úteis(!) mas é possível extender este mecanismo.

Identificação

Um protocolo de identificação de um agente *A* perante um agente *B* é

uma prova apresentada a B de que A conhece um segredo s sem que s possa vir a ser, em qualquer momento, conhecido por B.

Neste protocolo a prova é representada por uma chave pública k e o segredo representado pelo *trapdoor* t correspondente.

Protocolo 3 :

Inicialização *TA* gera, como no protocolo 1, chaves k e t e distribui t a *A*.

Intensão *A* manifesta a intensão de ser identificado publicando k

Desafio *B* gera um desafio σ no seguinte esquema:

(1) gera uma string aleatória ω ,

(2) calcula $\sigma \leftarrow k(\omega)$ e publicita-o.

Resposta *A* gera a resposta adequada ao desafio calculando $r \leftarrow t(\sigma)$ e publicando este valor.

Verificação *B* aceita a identificação sse $r = h(\omega)$.

Acordo de chaves com autenticação mútua

Dois agentes A e B, que possuem (cada um) um segredo na forma de trapdoors t_A *e* t_B*, sendo públicas as respectivas chaves* k_A *e* k_B*, acordam num segredo comum* λ *e, simultaneamente, certificam-se que estão a comunicar com o interlocutor legítimo.*

O protocolo tem duas partes: primeiro cada um dos participantes gera uma chave, λ_A e λ_B que, em princípio, coincidem. Na segunda parte verifica-se que as duas chaves coincidem e que cada agente está a interagir com o agente certo.

1ª Parte - Geração da Chave

Protocolo 4 :

segredo e desafio *A* gera o segredo comum λ com o esquema:

(1) gera ω aleatório e calcula $\lambda_A \leftarrow h(\omega)$

(2) calcula e publica $\sigma_1 \leftarrow k_B(\omega)$.

assume λ_A *como chave acordada*

resposta e desafio *B* calcula $\lambda_B \leftarrow t_B(\sigma_1)$

assume λ_B *como chave acordada*

▷ se a execução for correcta $\lambda_B = t_B(k_B(\omega)) = h(\omega) = \lambda_A$

2ª Parte - Autenticação da Chave

Protocolo 5 :

Desafio *B* calcula $\sigma_2 \leftarrow k_A(\lambda_B)$ e publicita-o.

Resposta *A* reconhece o interlocutor e responde ao desafio seguindo o esquema

(1) calcula $\lambda_1 \leftarrow t_A(\sigma_2)$

(2) *aceita prosseguir se* $\lambda_1 = h(\lambda_A)$

▷ se o protocolo for bem executado será $\lambda_1 = t_A(k_A(\lambda_B)) = h(\lambda_B) = h(\lambda_A)$

(3) calcula $\sigma_3 \leftarrow k_B(\lambda_1)$ e publicita-o.

Verificação *B* reconhece o interlocutor usando o esquema

(1) calcula $\lambda_2 \leftarrow t_B(\sigma_3)$

(2) *aceita prosseguir se* $\lambda_2 = h(h(\lambda_B))$

▷ se o protocolo for bem executado, será $\lambda_2 = t_B(k_B(\lambda_1)) = h(\lambda_1) = h(h(\lambda_B))$.

(3) *Termina com sucesso*

5.3 Grupos Diffie-Hellman

Em 1976, foi publicada pela primeira vez uma técnica criptográfica de chave pública: o protocolo de acordo de chaves que passou a ser designado por Diffie-Hellman.

Resumidamente o protocolo é formado por uma sequência de mensagens entre dois agentes, *A* e *B*, usando um canal aberto e com o objectivo de ambos partilharem um segredo λ.

Previamente os agentes concordam um primo p e num inteiro $g > 1$ cuja ordem[35] seja um primo r. Estas escolhas fixam um grupo multiplicativo $\mathbb{G}_g = \langle G, *, 1 \rangle$, de ordem r, em que o suporte G é o conjunto dos inteiros da forma $(g^n \mod p)$, com $n \in \mathbb{Z}_r$, e a operação de grupo $*$ é multiplicação módulo p.

Protocolo 6 : Diffie-Hellman

1. *A* gera um segredo aleatório $a \in \mathbb{Z}_r^*$, calcula g^a e envia este valor para *B*.
2. *B* gera um segredo aleatório $b \in \mathbb{Z}_r^*$, calcula g^b e envia este valor para *A*.
3. *A* calcula $(g^b)^a$; *B* calcula $(g^a)^b$; os valores coincidem e definem λ.

[35] A ordem de um qualquer $g \neq 0$ no grupo multiplicativo $\mathbb{Z}_p^*$ é o menor inteiro $n > 0$ tal que $(g^n = 1 \mod p)$. Todas as possíveis ordens de eventuais g são divisores de $(p-1)$.

A *correcção* do protocolo assenta nas propriedades algébricas dos sub-grupos cíclicos de $\mathbb{Z}_p^*$. O gerador g determina um destes sub-grupos, nomeadamente o grupo $\mathbb{G}_g$.

A *segurança* assenta na eventual intratabilidade do problema do logaritmo discreto ("discret-log problem" ou DLP) no grupo $\mathbb{G}_g$: se, no protocolo DH, for intratável recuperar a ou b a partir das mensagens g^a ou g^b, um atacante não consegue reconstituir o segredo λ.

Desde esse trabalho pioneiro, muitas mais técnicas criptográficas, com objectivos de segurança diversos, foram desenvolvidas assentes na mesma base: a correcção justificada pelas propriedades dos grupos cíclicos de inteiros e a segurança justificada pela intratabilidade do DLP nesses grupos.

No entanto cedo se notou que uma técnica criptográfica, baseada num determinado grupo cíclico, pode ser transferida para qualquer outro grupo cíclico desde que as condições de segurança se mantenham; isto é, o DLP continue intratável nesse grupo.

Sendo assim faz sentido considerar grupos multiplicativos arbitrários $\mathbb{G} = \langle G, *, 1 \rangle$. Note-se que pode ser importante considerar grupos infinitos.

Neste grupo os *elementos de torção* são aqueles $g \in \mathbb{G}$ para os quais existe um $n > 0$ tal que $g^n = 1$. O menor de todos estes n chama-se **ordem** de g. Os elementos de $\mathbb{G}$ da forma g^n formam um sub-grupo cíclico, cuja ordem é a ordem de g, e que se chama **órbita** de g.

Num tal grupo, o problema do logaritmo discreto, representado por $\mathrm{DLP}(\mathbb{G})$, define-se como

Problema do Logaritmo Discreto (DLP)
Conhecidos um elemento de torção $g \in \mathbb{G}$, a sua ordem r e um elemento g^a da sua órbita, determinar o valor de a.

Sabe-se que a complexidade computacional média do DLP num grupo cíclico é determinada pela dimensão do maior factor primo da ordem do grupo. Por isso, em tecnicas criptográficas assentes nos grupo cíclicos, *só interessa usar grupos cuja ordem seja um número primo*.

Se a ordem não for primo, um grupo com muitos elementos apenas introduz complexidade excessiva nas operações que devem ser eficientes sem que tal conduza a mais segurança para as operações supostamente intratáveis.

122 Definição
Os sub-grupos cíclicos de $\mathbb{G}$ cuja ordem é um primo, designam-se por **grupos primos** *em $\mathbb{G}$. A* **dimensão** *de $\mathbb{G}$, representada por $\|\mathbb{G}\|$, é $\lceil \log_2 r \rceil$, sendo r a ordem do maior grupo primo de $\mathbb{G}$.*

□

Simultaneamente, com o desenvolvimento de técnicas criptográficas com segurança assente no DLP, foi sendo claro que em certos grupos cíclicos era possível desenvolver outro tipo de técnicas. refinando as condições de segurança. Indo além da simples exigência de que o DLP seja intratável, introduziu-se uma gama mais detalhada de problemas

semelhantes ao DLP (mas que não coincidem necessariamente com o DLP) e introduziu-se a noção de "gap" para comparar a complexidade computacional destes vários problemas.

Isto permite explicitar formas mais complexas de segurança que, por seu lado, permitem definir técnicas com uma crescente gama de objectivos.

123 DEFINIÇÃO
Genericamente, diz-se que existe um **gap** *do problema P para o problema P' quando P é redutível a P' (isto é, existe um algoritmo PPT que converte qualquer eventual solução de P' numa solução de P) mas P' não é redutível a P.*

Quando existe um algoritmo PPT que resolve o problema P mas não existe um algoritmo PPT que resolva o problema P', este "gap" pode ser explorado para definir alguma forma de "trapdoor".

Repare-se no problema que um eventual atacante ao protocolo Diffie-Hellman quer resolver: ele conhece o gerador g, conhece as mensagens g^a e g^b e quer descobrir $\lambda = (g^a)^b = (g^b)^a = g^{a\,b}$.

Este problema é tão importante que merece uma designação própria: chama-se

Problema da Computação Diffie-Hellman (CDHP)
Conhecidos um elemento de torção g, a sua ordem r (supostamente um primo), e elementos g^a e g^b da sua órbita, determinar $g^{a\,b}$.

Note-se que para atacar o protocolo DH só é necessário resolver o CDHP; o ataque pode não exigir que se tenha de resolver DLP.

Portanto, como se comparam os dois problemas, DLP e CDHP?

Claramente, em qualquer grupo $\mathbb{G}$, CDHP é redutível a DLP: se se souber calcular a e b a partir de g^a e g^b, sabe-se facilmente calcular $g^{a\,b}$ a partir destes dois valores.

Já não é evidente que o DLP não seja redutível ao CDHP nem é evidente que, sendo o DLP computacionalmente intratável, o CDHP continue a ser computacionalmente intratável. Ambas as implicações dependem do grupo $\mathbb{G}$ usado[36].

Por isso faz sentido destacar os grupos onde seja seguro usar o protocolo DH,

124 Definição
Seja $\mathbb{G}$ um grupo multiplicativo e $g \in \mathbb{G}$ um elemento de torsão cuja ordem r é um número primo. O triplo $\langle \mathbb{G}, g, r \rangle$ designa-se por ***Grupo Diffie-Hellman (GDH)*** *se não existe nenhum algoritmo polinomial em $|r|$ que resolva CDHP.*

Num GDH $\langle \mathbb{G}, g, r \rangle$ faz sentido definir os problemas DLP e CDHP como oráculos. Assim define-se

[36] Num grupo primo de ordem r, sabe-se que DLP e CDHP são equivalente quando existe um grupo auxiliar definido algebricamente sobre $\mathbb{Z}_r$ que tenha pequena dimensão. Os grupos auxiliares que têm sido mais testados são as curvas elípticas definidos sobre $\mathbb{Z}_r$.

Oráculo DLP Idealiza um algoritmo que recebe como *input* um $x \in G$ e produz como resultado o único $a \in \mathbb{Z}_r$ tal que $x = g^a$ ou então falha se x não está na órbita de g.

Oráculo CDHP Idealiza o algoritmo que recebe como *input* $\langle x, y \rangle \in G^2$ e falha se x ou y não pertencem à órbita de g; se for $x = g^a$ e $y = g^b$ o oráculo produz o resultado $z = g^{a\,b}$.

O problema da computação Diffie-Hellman pode-se generalizar para dois grupos DH com a mesma ordem r. Considere-se um segundo GDH $\langle \Gamma, \gamma, r \rangle$ de ordem r. Então, conhecidos ambos GDH´s,

Problema da Computação Diffie-Hellman Generalizada (co-CDHP)
Dado $g^a \in G$ *determinar* $\gamma^a \in \Gamma$

É óbvio que co-CDHP generaliza o CDHP: basta fazer $\Gamma = G$ e $\gamma = g^b$. É também óbvio que co-CDHP é redutível a DLP: consultando uma vez um oráculo DLP, dado $x \in G$ obtém-se o valor a que permite calcular γ^a.

O oráculo respectivo define-se do mesmo modo

Oráculo co-CDHP Idealiza o algoritmo que recebe $x \in G$, falha se x não está na órbita de g, e, caso seja $x = g^a$, produz γ^a.

□

Dado um GDH $\langle \mathbb{G}, g, r \rangle$ que técnicas criptográficas é possível definir?

Como primeira aplicação criptográfica de grupos DH temos a construção de um **sistema trapdoor** tal como foi sugerido na secção 1. Daí resultam um conjunto de técnicas básicas como foi visto nessa altura.

No entanto outras técnicas criptográficas usam esta estrutura básica dos grupos DH explorando as propriedades algébricas do grupo sem que possam ser expressas num enquadramento trapdoor genérico.

Dentro destas destacamos as técnicas criptográficas afins ao esquema de assinaturas digitais **DSA** e as técnicas criptográficas ditas **orientadas à identidade**.

Protocolo 7 : Componentes Comuns

Sejam

1. $\langle \mathbb{G}, g, r \rangle$ um grupo Diffie-Hellman; G denota a órbita de g.
2. $H : \mathbb{B}^* \longrightarrow \mathbb{Z}_r^*$ é uma função de hash
3. $f : G \longrightarrow \mathbb{Z}_r^*$ é uma função chamada *função de redução*.

As técnicas criptográficas baseadas na identidade, para além de grupos DH com características particulares, usam outro tipo de componentes.

Técnicas básicas trapdoor num grupo DH

A partir das componentes comuns define-se um *sistema trapdoor* com

- A *função de hash*: $h : \mathbb{B}^* \longrightarrow G$ define-se como

$$h \ : \ x \ \mapsto \ g^{H(x)}$$

- O *sistema de chaves públicas*: para cada $\beta \neq 1 \in G$ define-se $k_\beta : \mathbb{B}^* \longrightarrow G$ como

$$k_\beta \ : \ x \ \mapsto \ \beta^{H(x)}$$

- O *sistema de chaves privadas*: para cada $a \in \mathbb{Z}_r^*$, define-se $t_a : G \longrightarrow G$ como

$$t_a \ : \ z \ \mapsto \ z^a$$

- A *função geradora*: $\mathcal{G} : \mathbb{Z}_r^* \longrightarrow G$

$$\mathcal{G} \ : \ a \ \mapsto \ g^{a^{-1}}$$

Este sistema verifica claramente a relação

$$\beta = \mathcal{G}(a) \quad \Leftrightarrow \quad t_a(k_\beta(x)) \ = \ h(x) \qquad \forall a, x \in \mathbb{Z}_r^*$$

e, por isso, é um presumível sistema trapdoor.

Nestas circunstâncias todas as técnicas criptográficas básicas descritas na secção 2 podem ser usadas. A título ilustrativo uma cifra assimétrica seria

Protocolo 8 : ElGamal Generalizado

geração do par de chaves

Gerar $u \in \mathbb{B}^*$ aleatório; calcular $a \leftarrow H(u)$ e $\beta \leftarrow \mathcal{G}(a)$; a chave privada é t_a ; a chave pública é k_β .

cifrar $x \in \mathbb{Z}_r$

Gerar $\upsilon \in \mathbb{B}^*$ aleatório; calcular a redundância $\sigma \leftarrow k_\beta(\upsilon)$, a imagem $y \leftarrow x \oplus f(h(\upsilon))$ e o criptograma $z \leftarrow \sigma \,\|\, y$. Publicitar z.

decifrar $z = \sigma \,\|\, y$

Recuperar $(\sigma, y) \leftarrow z$. Recuperar $x' \leftarrow y \oplus f(t_a(\sigma))$.

Correcção As variáveis x e x' são indistinguíveis.

Segurança A família de funcionais $C_a \colon \mathbb{Z}_r \to \wp(\mathbb{Z}_r \times \mathbb{Z}_r)$, indexada por $a \in \mathbb{Z}_r^*$ e definida por $C_a(x) = \{\, \sigma \| y \mid \upsilon \in \mathbb{B}^* \,\}$, com $\sigma = k_{\mathcal{G}(a)}(\upsilon)$ e $y = x \oplus f(h(\upsilon))$, é um sistema de chaves.

A condição de segurança significa que todas as funcionais são unidireccionais e não é possível recuperar a mesmo que se conheça o texto claro x e o criptograma $\sigma \,\|\, y$.

5.4 OAE - Optimal Asymetric Encryption

Muitas cifras assimétricas (e todas as simétricas) são *determinísticas*; isto é, cifrar duas vezes o mesmo texto produz duas vezes o mesmo criptograma. Na construção de uma cifra é frequentemente necessário introduzir um mecanismo de "randomização" transformando a cifra determinística numa cifra não-determinística. Um desses mecanismos é o que a seguir se aparesenta.

Protocolo 9 : OAE - Randomização de uma cifra k

Setup São escolhidas duas funções de "hash" $G : \mathbb{B}^m \longrightarrow \mathbb{B}^n$ e $H : \mathbb{B}^n \longrightarrow \mathbb{B}^m$; é dada uma cifra determinística k como uma permutação em $\mathbb{B}^{n+m}$.

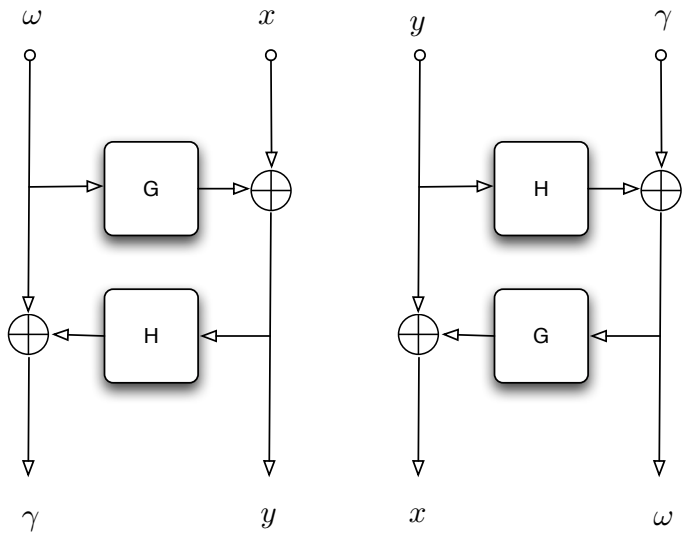
Cifra Dado um texto x, obter uma codificação "randomizada" $y\|\gamma$ e um criptograma σ

(1) $\omega \leftarrow \mathbb{B}^m \quad ; \quad y \leftarrow x \oplus G(\omega) \quad ; \quad \gamma \leftarrow \omega \oplus H(y)$
(2) $. \longrightarrow \sigma = k(y\|\gamma)$

Decifra Dado o criptograma σ recuperar a codificação $y\|\gamma$ e depois o texto x

(1) $y\|\gamma \leftarrow k^{-1}(\sigma)$
(2) $\omega \leftarrow H(y) \oplus \gamma \quad ; \quad x \leftarrow y \oplus G(\omega)$

O custo desta "randomização" está no facto de a cifra k, que é uma permutação de $n+m$ bits, é usada para cifra mensagens de n bits. A randomização usa os outros m bits.



Cada para de funções de "hash" G, H determina uma codificação $\mathcal{C}^{G,H} : x\|\omega \mapsto y\|\gamma$ e uma descodificação $\mathcal{D}^{G,H} : y\|\gamma \mapsto x\|\omega$.

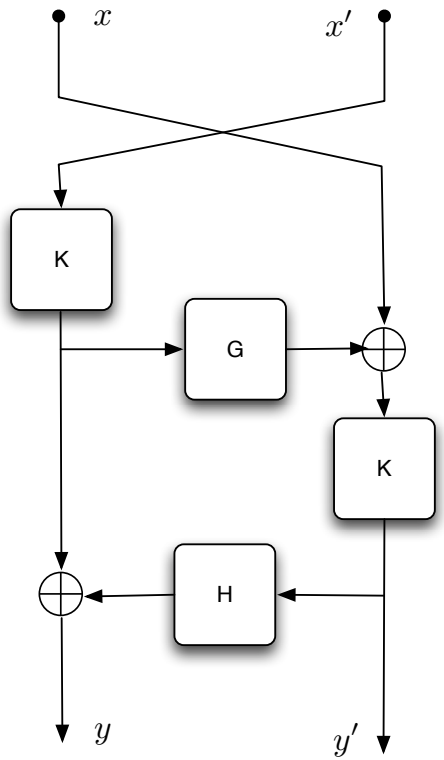
A figura indica o esquema de codificação e o esquema de descodificação e, é claro, que estas duas transformações são muito semelhantes.
De facto, recorrendo ao operador troca $t : (a, b) \mapsto (b, a)$, temos $\mathcal{D}^{G,H} = t^{-1} \cdot \mathcal{C}^{H,G} \cdot t$.

Este par de esquemas designa-se por "Optimal Asymetric Encryption". A referência a "asymetric encryption" deriva do seu objectivo quando foi primeiro introduzido por Boneh: "randomizar" a cifra RSA que, devido ao facto de preservar o produto de textos[37] potencia vários tipos de ataques.

A designação "Optimal" deriva do facto de ser possível construir aproximações que conduzam a provas probabilísticas

[37] No RSA cifrar um texto $x = x_1\, x_2$ resulta em $(x_1\, x_2)^k = x_1^k\, x_2^k$.

de segurança.



Esta arquitectura pode também ser usada como aperfeiçoamento de uma cifra por blocos K duplicando o tamanho do bloco e melhorando a imunidade contra ataques algébricos. A figura à esquerda ilustra esta abordagem:

1. O bloco "input" é dividido em duas componentes $x||x'$. Eventualmente uma das componentes pode conter uma sub-componente aleatória de modo a introduzir randomização nestas cifra. De forma semelhante o bloco "output" é dividido nas componentes $y||y'$.
2. As funções de "hash" G e H são substituídas por S-boxes (eventualmente numa construção "bricklayer" para permitir uma implementação eficiente) com não-linearidade suficiente para evitar as bileaniarides que conduzem a ataques algébricos.
3. Os blocos K denotam o mesmo esquema de cifra mas com duas chaves diferentes. Cada um deles actua sobre metade do bloco, mas a arquitectura OAE fornece uma mistura adicional das chaves.

5.5 O esquema de assinaturas DSA generalizado

Dentro das técnicas criptográficas mais importantes construídas com grupos DH estão, sem dúvida, os esquemas de assinaturas digitais que nasceram do **Digital Signature Algorithm (DSA)**.

O DSA foi proposto em 1991 pelo "National Institute of Standards and Tecnhology (NIST)" e foi aceite em 1994 como "standard" de assinatura digitais pelo organismo de standardização NIST. Conjuntamente com a função de hash SHA ("Standard Hash Algorithm")[38] constituem o "Digital Signature Standadrd" (DSS).

O DSA é uma técnica baseada nas propriedades algébricas do mais comum dos grupos cíclicos criptográficos: o grupo $\mathbb{Z}_p^*$. Com o aparecimento de algoritmos eficientes para lidar com curvas elípticas sobre corpos finitos, foi proposto em 1992 o **Elliptic Curve Digital Signature Algorithm (ECDSA)** como adaptação para o grupo cíclico daí resultante.

De momento o ECDSA é aceite como standard ISO (ISO 14888-3) desde 1998, como standard ANSI (ANSI X9.62) desde 1999 e, desde 2000, como standard comum IEEE e FIPS (IEEE 1363-2000 e FIPS 186-2).

A versão aqui apresentada é geral e usa um qualquer GDH e as componentes comuns apresentadas na página 309.

[38] Posteriormente substituído pelo SHA-1.

Protocolo 10 : DSA Generalizado

geração do par de chaves

1. Gerar $u \in \mathbb{B}^*$ aleatório e calcular $a \leftarrow H(u)$ e $\beta \leftarrow h(u)$.
2. A chave privada é a e a chave pública é β.

assinar a mensagem $m \in \mathbb{B}^*$

1. Gerar $\upsilon \in \mathbb{B}^*$ aleatório; calcular $\gamma \leftarrow h(\upsilon)$ e $\sigma \leftarrow f(\gamma)$.
2. Determinar s como solução da equação $s\,H(\upsilon) = a\,\sigma + H(m)$ em $\mathbb{Z}_r$.
3. Publicitar a assinatura $z \leftarrow \sigma \,\|\, s$

verificar a assinatura z para o texto m

1. Recuperar $(\sigma, s) \leftarrow z$; calcular $\lambda \leftarrow (\beta^{\sigma} \cdot h(m))^{s^{-1}}$
2. Aceitar a assinatura sse $\sigma = f(\lambda)$ se verifica em $\mathbb{Z}_r$,

As várias instâncias deste esquema genérico resultam de se fazer uma escolha apropriada do grupo cíclio $\mathbb{G}$, da função de hash H e da função de redução f. As duas instâncias mais importantes são o DSA original e o ECDSA que correspondem às seguintes escolhas:

DSA

O grupo cíclio é um grupo primo do grupo multiplicativo $\mathbb{Z}_p^*$ cuja ordem r divide $p-1$; o standard DSA exige que a dimensão de r seja ≥ 160 bits e a dimensão de p seja ≥ 512 bits.

A função de redução $f : G \longrightarrow \mathbb{Z}_r$ é $x \mapsto x \mod r$. A função de hash H é a função SHA-1 com os resuldados reduzidos módulo r.

ECDSA

Os elementos de um grupo cíclico definido por uma curva elíptica são (como veremos adiante) pontos $P = \langle x, y \rangle$ do plano em que as coordenadas x, y são elementos de um determinado corpo finito $\mathbb{F}_q$. Sobre estes pontos está definida uma operação de grupo que, neste caso, é representado de forma aditiva; isto é $P + Q$ é a aplicação da operação de grupo a dois pontos P e Q e existe um elemento neutro (um zero) representado por $\mathcal{O}$.

Como a operação de grupo é aditiva, a "exponenciação" é aqui chamada **multiplicação escalar** e representa-se por $n\,P$. A *ordem* de P é o menor n tal que $n\,P = \mathcal{O}$.

O ECDSA usa, como grupo cíclico G, a órbita de um ponto P de ordem prima r. A função de hash H é, como no DSA, a função SHA-1 com os resultados reduzidos módulo r.

A função redução f mapeia um ponto $P \in G$ de coordenadas $\langle x, y\rangle \in \mathbb{F}_q \times \mathbb{F}_q$ num inteiro $f(P) \in \mathbb{Z}_r$ que é a redução módulo r de uma representação inteira[39] da componente x do ponto P.

□

Sumariamente a correcção da assinatura deriva da seguinte cadeia de argumentos.

1. A equação

$$s\,H(v) = a\,\sigma + H(m) \tag{114}$$

verifica-se em $\mathbb{Z}_r$ sse $g^{s\,H(v)} = g^{a\,\sigma} \cdot g^{H(m)}$ se verifica em G.

2. Como $\beta = g^a$, $\gamma = g^{H(v)}$ e $h(m) = g^{H(m)}$, a equação (114) verifica-se sse $\gamma^s = \beta^\sigma \cdot h(m)$.
3. Como $\lambda = (\beta^\sigma \cdot h(m))^{s^{-1}} = (\gamma^s)^{s^{-1}} = \gamma$ e $\sigma = f(\gamma)$, a assinatura será aceite sse for $\sigma = f(\lambda)$.

Este argumento de **correcção** apenas prova que, caso a assinatura esteja bem construída, a verificação tem sucesso. Falta provar a implicação em sentido contrário: assume-se que a a verificação tem sucesso e quere-se provar que a assinatura foi bem construída.

[39] Se $\mathbb{F}_q$ for um corpo primo, a representação inteira de $x \in \mathbb{F}_q$ coincide com x; se $\mathbb{F}_q$ for um corpo binário, a representação inteira de x será o inteiro que tem a mesma representação em bits do que x.

A noção de grupo cíclico implica que uma equação da forma $g^x = g^y$ é válida em G se e só se $x = y$ for válida em $\mathbb{Z}_r$. Portanto temos a certeza que se verifica $\gamma^s = \beta^\sigma \cdot h(m)$ se e só se a equação (114) se verifica.

Porém a intervenção da função de redução f neste esquema, traz problemas.

Note-se que, se f não for injectiva, é possível ocorrer $f(\gamma) = f(\gamma')$ com $\gamma \neq \gamma'$. Portanto a verificação de $f(\gamma) = f(\lambda)$ não implica necessariamente que seja válido $\gamma^s = \beta^\sigma \cdot h(m)$.

Se facto, se percorrermps todos os possíveis $\gamma_1 \in f^{-1}(\sigma)$ e os respectivos b_1 tais que $\gamma_1 = g^{b_1}$, é possível que surjam vários tipos de situações:

1. existam mensagens m_1 cujo hash $H(m_1)$ verifique a equação $s\,b_1 = a\,\sigma + H(m_1)$; assim, surgem mensagens diferentes (m e m_1) com *hashs* diferentes que verificam a mesma assinatura s.
2. existam assinaturas $s_1 \neq s$ que verifiquem $s_1\,b_1 = s\,b$; neste caso, o mesmo $\lambda \leftarrow \beta^\sigma\,h(m)$, agora levantado a outro expoente ${s_1}^{-1}$, continuaria a verificar $\sigma = f\left(\lambda^{{s_1}^{-1}}\right)$; temos, então, duas assinaturas diferentes (s e s_1) que o esquema de verificação aceita para a mesma mensagem m.

Estas duas situações indicam-nos que, para um mesmo σ (e uma mesma geração aleatória υ) podem existir várias assinaturas para a mesma mensagem e várias mensagens com a mesma assinatura. Portanto os pares

mensagem+assinatura legitimamente gerados pelo esquema de assinatura não coincidem exactamente os pares análogos que são verificados pelo esquema de verificação.

No entanto as construções apresentadas indicam que não será probabilisticamente viável distinguir a duas situações; para fazer tal distinção não só seria necessário inverter a função de hash como resolver o problema do logaritmo discreto para, dado $\gamma \in f^{-1}(\sigma)$, encontrar o valor b tal que $\gamma = g^b$.

□

Para uma análise da **segurança**, e sob o ponto de vista de um atacante (aqui designado por **falsificador**), vamos assumir que ele:

(i) conhece o grupo cíclico $\langle \mathbb{G}, g, r \rangle$,
(ii) escolhe um assinante legítimo seleccionando a sua chave pública β, e
(iii) escolhe também uma uma família finita de textos $M = \{m_i\}$ e conhece assinaturas para cada um desses textos; isto é, conhece $S = \{\langle \sigma_i, s_i \rangle\}$ com $(m_i, \langle \sigma_i, s_i \rangle) \in \mathrm{Ver}_\beta$.

Colocam-se-lhe dois tipos de desafios/ataques:

falsificação da mensagem
: Escolher um dos textos em M, digamos $m_i \in M$, e gerar um outro texto $m' \neq m_i$ que verifique a mesma assinatura $\langle \sigma_i, s_i \rangle$ com a mesma chave pública β.

falsificação da assinatura

Gerar um qualquer texto $m \notin M$ e uma assinatura $\langle \sigma, s \rangle$ que seja verificável com a chave pública do assinante legítimo.

Uma análise *ad-hoc* dos ataques começa por calcular a família $\Gamma = \{\gamma_i\}$ fazendo $\lambda_i \leftarrow \beta^{\sigma_i} h(m_i)$ e resolvendo $\gamma_i^{s_i} = \lambda_i$.

Para o primeiro ataque ("falsificação da mensagem"), a forma mais simples seria encontrar uma mensagem m' que verifique $H(m') = H(m_i)$ para algum dos $m_i \in M$. Isto equivale a encontrar colisões na função de *hash* H.

Assumindo que não existem tais colisões, a alternativa passa por encontrar um i e um $\gamma' \neq \gamma_i$ tais que $f(\gamma') = f(\gamma_i) = \sigma_i$. Então faz-se $h(m') \leftarrow \beta^{-\sigma_i} \left(\gamma'\right)^{s_i}$ e consegue-se localizar uma *hash* $h(m') \neq h(m_i)$ que é aceite pelo esquema de verificação.

Esta computação é realizável por um algoritmos PPT. Será possível, a partir de $h(m')$, determinar a desejada mensagem m' também por um algoritmo PPT ?

Sabendo que $h(m') = g^{H(m')}$, se fosse possível determinar m', seria também possível resolver o problema do logaritmo discreto para o valor $h(m')$: bastaria calcular $H(m')$. Se $h(m')$ for "suficientemente aleatório", essa solução para o PLD é, por hipótese, intratável.

Uma possível realização ao ataque de "falsificação de assinatura" parte, simplesmente, do conhecimento da chave privada a; se for possível determinar a a partir de $\beta = g^a$ (resolvendo o problema do logaritmo discreto) seria possível, obviamente, determinar qualquer assinatura para qualquer mensagem.

Uma questão bastante mais complexa é a de saber se é possível falsificar a assinatura de uma mensagem $m \notin M$ sem o conhecimento da chave privada a.

5.6 Transformação de Fiat-Samir

A transformação de Fiat-Shamir converte um qualquer protocolo de identificação "desafio-resposta" na forma canónica, num esquema de assinaturas.

O protocolo de identificação lida com dois agentes, normalmente designados por "prover" (representado por $\mathcal{P}$) e "verifier" (representado por $\mathcal{V}$). O seu objectivo é fazer com que o "prover" prove ao "verifier" que conhece um segredo s sem que, neste processo, o "verifier" fique com qualquer conhecimento sobre esse segredo. Nomeadamente o "verifier" só deve ser capaz, no fim do protocolo, de conhecer exactamente os mesmo items de informação que conheceria se não participasse no protocolo.

Neste e nos seguintes protocolos usaremos a seguinte convenção:

- A notação $A : x \leftarrow \phi$ representa um passo de protocolo em que o agente A fica a conhecer um item x a partir de uma computação ϕ.
- A notação $A : x \rightarrow \psi$ denota um passo de protocolo onde o agente A, a partir do conhecimento x, calcula um valor ψ e publicita-o. A notação $A : x \rightarrow y = \psi$ é semelhante mas atribui o nome y à informação calculada.
- O titular público é representado por $\cdot$ e o conhecimento vazio por $*$.

De uma forma esquemática o protocolo canónico "desafio-resposta" escreve-se

Protocolo 11 : Identificação Canónica

$\mathcal{P}: * \longrightarrow \text{id}$	$\mathcal{P}$ mostra **intensão** de ser identificado tornando público id
$\mathcal{V}: * \longrightarrow d$	$\mathcal{V}$ gera um **desafio** aleatório d com um grau de incerteza adequado
$\mathcal{P}: s, d \longrightarrow r$	$\mathcal{P}$ usa o segredo s e o desafio d para calcular uma **resposta** r
$\mathcal{V}: \rho(\text{id}, d, r) =? 1$	$\mathcal{V}$ **verifica**, recorrendo a uma decisão ρ, se é válido o triplo $\langle \text{id}, d, r \rangle$

Assim, uma instância específica deste protocolo necessita de precisar os três algoritmos e a decisão que nele intervêm. Nomeadamente:

1. O algoritmo PPT que, a partir da identidade de $\mathcal{P}$, gera um valor id representativo;
2. O gerador de desafios aleatórios d;
3. O algoritmo PPT que, sob *input* do segredo s e do desafio d, produz a resposta adequada r
4. A decisão ρ que reconhece os triplos $\langle \text{id}, d, r \rangle$ válidos.

A decisão ρ é um teste raro que diferencia duas linguagens: a linguagem dos triplos $\langle \text{id}, d, r \rangle$ válidos gerada fazendo d percorrer o domínio dos desafios aceitáveis, e a linguagem dos peseudo-triplos $\langle \text{id}, d, u \rangle$, quando u é aleatório no domínio das respostas aceitáveis.

EXEMPLO 31: Um exemplo paradigmático de um protocolo com esta estrutura assente em grupos Diffie-Hellman é o chamado **protocolo de identificação de Schnorr**.

Nesse protocolo vamos considerar os elementos comuns de um grupo DH (ver página 309) e ainda um gerador $\mathcal{U}_r$ de inteiros uniformemente distribuídos em $\mathbb{Z}_r$.

Protocolo 12 : Identificação de Schnorr

SetUp geração de chaves, privada s e pública $\beta = g^{-s}$

(1) $\mathcal{P}: s \leftarrow \mathcal{U}_r \quad ; \quad \mathcal{P}: s \rightarrow \beta = g^{-s}$

Instância o "prover" $\mathcal{P}$ e o "verifier" $\mathcal{V}$ executam os seguintes passos:

(1)	$\mathcal{P}: \upsilon \leftarrow \mathcal{U}_r \quad ; \quad \mathcal{P}: \upsilon \rightarrow \gamma = g^{\upsilon}$	*intensão*
(2)	$\mathcal{V}: d \leftarrow \mathcal{U}_r \quad ; \quad \mathcal{V}: d \rightarrow d$	*desafio*
(3)	$\mathcal{P}: \upsilon, s, d \rightarrow r = \upsilon + s\,d$	*resposta*
(4)	$\mathcal{V}: g^r\,\beta^d \stackrel{?}{=} \gamma$	*verificação*

Note-se que, sendo o protocolo cumprido, $r - s\,d = \upsilon$ e, por isso, $g^r\,(g^{-s})^d = g^{\upsilon}$ o que conduz a $g^r\,\beta^d = \gamma$.

Considere-se de novo a estrutura geral do protocolo de identificação (pag. 324) e as componentes a ele associadas.

A transformação de Fiat-Shamir acrescenta a este "setup" uma função de *hash* H e a operação concatenação de códigos representadas pelo operador $\|$. Implementa um esquema de assinaturas da seguinte forma:

Protocolo 13 : Assinatura de Fiat-Shamir

Assinatura O "prover" $\mathcal{P}$, titular do segredo s, produz a assinatura σ para a mensagem m

(1) $\mathcal{P}: d \leftarrow H(\mathsf{id} \| m)$
(2) $\mathcal{P}: d, s \rightarrow \sigma = \mathsf{id} \| r(s, d)$

Verificação O "verifier" $\mathcal{V}$ verifica a validade de $\sigma = \mathsf{id} \| r$ como assinatura de m.

(1) $\mathcal{V}: \mathsf{id}, r \leftarrow \sigma$
(2) $\mathcal{V}: d \leftarrow H(\mathsf{id} \| m)$
(3) $\mathcal{V}: \rho(\mathsf{id}, d, r) =? 1$

5.7 Assinaturas com recuperação de mensagem e o esquema de assinaturas Hermes

O esquema de assinaturas Hermes é um projecto público do Ministério da Defesa de França para um esquema de assinaturas com recuperação de mensagens, assente em grupos Diffie-Hellman, que segue a estrutura de uma transformação de Fiat-Shamir sobre um protocolo de identificação análogo ao protocolo de Schnorr.

Para caracterizar Hermes temos de olhar separadeamente para as suas duas componentes essenciais: o que é um assinatura com recuperação de mensagens e qual é o protocolo "desafio-resposta" a que se vai aplicar a transformação FS.

Assinaturas com recuperação de mensagem (ARN)

Designemos por $\mathcal{M}$ o espaço das mensagens e por $\mathcal{S}$ o espaço das assinaturas.

Quando uma mensagem $m \in \mathcal{M}$ é assinada digitalmente e enviada para um destinatário é necessário enviar não só a assinatura construída $\sigma \in \mathcal{S}$ mas também o texto da mensagem m.

Por vezes a estrutura das mensagens é tal que é possível ter uma solução mais eficiente (em termos de bits transmitidos) num esquema onde a própria assinatura contenha informação sobre a mensagem.

Um tal esquema é formado pelas seguintes componentes:

Função de redundância É uma função $\xi : \mathcal{M} \rightarrow \mathbb{N}$ com as seguintes proprieades:

1. ξ é uma função injectiva implementável por um algoritmo PPT determinístico.
2. É implementável PPT a função parcial ξ^{-1} que verifica $\xi^{-1}(u) = x$, quando $u = \xi(x)$, e $\xi^{-1}(u) = \perp$ quando $u \notin \xi(\mathcal{M})$.
3. ξ "espalha" os seus resultados "uniformemente" por todo $\mathbb{N}$. Formalmente, existe um $l > 0$ suficientemente grande tal que, para todo $n > 0$, $|\xi(\mathcal{M}) \cap \mathbb{B}^n|/2^n \leq 2^{-l}$.

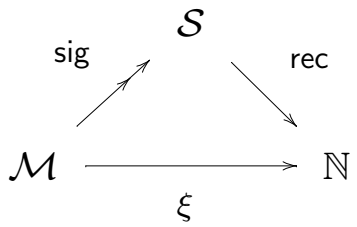
Funcional de assinatura

$\text{sig}_s : \mathcal{M} \rightarrow \wp(\mathcal{S})$, é uma funcional que depende de uma chave privada s, e é unidireccional.

Função de recuperação

$\text{rec}_\beta : \mathcal{S} \rightarrow \mathbb{N}$ é uma função que depende da chave pública β. e que verifica a condição de correcção

$$\sigma \in \text{sig}_s(m) \quad \Leftrightarrow \quad \xi(m) = \text{rec}_\beta(\sigma) \tag{115}$$



Num tal esquema a assinatura σ é aceite quando de verifica

$$\xi^{-1}\left(\text{rec}_\beta(\sigma)\right) \neq \perp \tag{116}$$

e, se for aceite, o valor de $\xi^{-1}\left(\text{rec}_\beta(\sigma)\right)$ recupera a mensagem m.

Esquema de assinaturas Hermes

O esquema de assinaturas HERMES actua sobre mensagems m decompostas em duas componentes, $m = m_1 \| m_2$ e só trata a primeira m_1 num esquema ARM.

Assume-se que ambos agentes conhecem a segunda parte m_2.

HERMES assume as seguintes componentes

- Um grupo de Diffie-Hellman $\langle \mathbb{G}, g, r \rangle$;
- Duas funções de *hash*: $H_1 : G \to \mathbb{Z}_t$, com $t \gg r$, e $H_2 : \mathbb{Z}_t \to \mathbb{Z}_r$;
- Uma função de redundância $\xi : \mathbb{Z}_r \to \mathbb{Z}_t$;
- Um gerador $\mathcal{U}_r$ de números aleatórios uniformemente distribuídos em $\mathbb{Z}_r$.

A correcção deste esquema é facilmente verificado; note-se que, com $e = -H_2(z\|m_2)$, se tem $g^x = g^\upsilon \beta^{-e}$. Portanto $H_1(g^\upsilon) = H_1(g^x \beta^e) = h$. Como $r = H_1(g^\upsilon) \oplus \xi(m_1) = h \oplus \xi(m_1)$ e como $\tau = h \oplus z$, temos de concluir que $\tau = \xi(m_1)$.

Protocolo 14 : Hermes: Assinatura com Recuperação de Mensagem

Geração de chaves $\mathcal{P}$ gera a chave privada s e a chave pública $\beta = g^s$

(1) $\mathcal{P}: s \leftarrow \mathcal{U}_r \quad ; \quad \mathcal{P}: s \rightarrow \beta = g^s$

Assinatura $\mathcal{P}$ gera uma assinatura ARM σ da componente m_1, usando m_2 como chave

(1) $\mathcal{P}: \upsilon \leftarrow \mathcal{U}_r \quad ; \quad \mathcal{P}: z \leftarrow H_1(g^{\upsilon}) \oplus \xi(m_1)$
(2) $\mathcal{P}: x \leftarrow \upsilon + s\, H_2(z \,\|\, m_2) \quad ; \quad \mathcal{P}: x, z \rightarrow \sigma = z \| x$

Recuperação $\mathcal{V}$ recupera, de σ e m_2, um valor $\tau = \xi(m_1)$

(1) $\mathcal{V}: z, x \leftarrow \sigma \quad ; \quad \mathcal{V}: e \leftarrow -H_2(z \,\|\, m_2)$
(2) $\mathcal{V}: h \leftarrow H_1(g^x\, \beta^e) \quad ; \quad \mathcal{V}: h, z \rightarrow \tau = h \oplus z$

Verificação $\mathcal{V}$ aceita a mensagem sse τ estiver na imagem $\xi(\mathbb{Z}_r)$,

(1) $\mathcal{V}: \xi^{-1}(\tau) \neq \perp$

O esquema Hermes pode também ser obtido aplicando a transformação de Fiat-Shamir à seguinte variante do

protocolo de identificação de Schnorr.

Protocolo 15 : Identificação Schnorr-Hermes

Geração de chaves como no protocolo 14

Instância

(1)	$\mathcal{P}: v, v' \leftarrow \mathcal{U}_r, \mathcal{U}_r \quad ; \quad \mathcal{P}: \tau \leftarrow \xi(v')$	
(2)	$\mathcal{P}: v, \tau \rightarrow z = H_1(g^v) \oplus \tau$	*intensão*
(3)	$\mathcal{V}: d \leftarrow \mathcal{U}_r \quad ; \quad \mathcal{V}: d \rightarrow d$	*desafio*
(4)	$\mathcal{P}: d, s, v \rightarrow r = v + s\, d$	*resposta*
(5)	$\mathcal{V}: \tau' \leftarrow H_1(g^r\, \beta^{-d}) \oplus z$	
(6)	$\mathcal{V}: \xi^{-1}(\tau') \neq \perp$	*verificação*

5.8 Assinaturas Cegas

Uma das aplicações mais interessantes e complexas da criptografia são os protocolos de votação electrónica. Sem querer apresentar aqui uma descrição completa deste tipo de técnicas, parece ser claro que dois dos seus objectivos nucleares são as garantias de *anonimato do voto* e de *autenticidade do votante*.

Se, por um lado, o votante deve ser autenticado por uma qualquer autoridade (a *mesa* de voto) também se deve garantir que essa autoridade não tem conhecimento do voto específico do votante. Quando a mesa reconhece a legitimidade do votante, deve autenticar o voto respectivo sem o conhecer; em seguida deve passar essa informação ao *escrutinador* que faz a contagem dos votos.

Este esboço de protocolo é suficiente para se perceber que a sua componentes fundamental é uma técnica de autenticação designada por **assinatura cega**.

Numa assinatura cega (conceito introduzido por CHAUM nos anos 80) existem dois agentes: o emissor $\mathcal{E}$ e o "prover" $\mathcal{P}$. O objectivo de $\mathcal{E}$ é obter uma assinatura de $\mathcal{P}$ sobre uma mensagem m por si escolhida, sem que $\mathcal{P}$ conheça essa mensagem.

Uma solução "aparentemente óbvia" consistiria em esconder a mensagem através de uma função de *hash* H: o emissor calcularia $x \leftarrow H(m)$ e fornecia ao "prover" apenas x; ao assinar x, $\mathcal{P}$ continuaria sem conhecer m.

O protocolo de votação electrónica ilustra imediatamente porque é que esta não é uma solução aceitável; se a mensagem m for um voto, dado que existe um número limitado de alternativas de voto, ao "prover" bastaria

percorrer todas essas alternativas até encontrar aquela que tivesse um "hash" igual a x; desta forma determinaria o voto. Portanto, *esconder m de $\mathcal{P}$* exige também esconder o respectivo *hash*.

O protocolo original de Chaum baseia-se nas assinaturas RSA.

Protocolo 16 : Assinatura Cega (Chaum)

"Setup" RSA O "prover" $\mathcal{P}$ gera um módulo RSA produto de dois primos, $n = p \cdot q$ e $\phi \leftarrow (p-1)(q-1)$. Gera a chave privada s e a respectiva chave pública k.

$$(1) \quad \mathcal{P}: s \leftarrow \mathbb{Z}_\phi^* \quad ; \quad s \rightarrow k = s^{-1} \pmod{\phi}$$

Assinatura $\mathcal{E}$ cifra o *hash* $x = H(m)$; $\mathcal{P}$ assina o criptograma respectivo; $\mathcal{E}$ recupera a assinatura e verifica-a.

$$(1) \quad \mathcal{E}: x \leftarrow H(m) \quad ; \quad \mu \leftarrow \mathbb{Z}_n^* \quad ; \quad \mu, h \rightarrow y = \mu^k \cdot h \pmod{n}$$
$$(2) \quad \mathcal{P}: s \rightarrow x = y^s \pmod{n}$$
$$(3) \quad \mathcal{E}: \sigma \leftarrow \mu^{-1} \cdot x \pmod{n} \quad ; \quad \sigma^k \pmod{n} \stackrel{?}{=} h$$

Correcção

Porque $y = \mu^k \cdot h$ tem-se $x = y^s = \mu^{k\,s} \cdot h^s = \mu \cdot h^s$ (pelo teorema RSA, $\mu^{k\,s} = \mu$); portanto, $\sigma = \mu^{-1} \cdot x = h^s$. Desta forma $\mathcal{E}$ recupera a assinatura h^s da mensagem M. O "prover" $\mathcal{P}$ conhece $\mu^k \cdot h$ e consegue calcular $\mu \cdot h^s$; nenhum destes valores permite-lhe determinar μ ou h.

Um segundo protocolo de assinaturas usa as componentes comuns dos grupos Diffie-Hellman apresentados no protocolo 7 (página 309).

Protocolo 17 : Assinatura Cega (Schnorr)

Setup O "prover" $\mathcal{P}$ escolhe uma instância de um grupo DH, publicita-o; gera uma chave privada e publicita a respectiva chave pública obtendo a sua autenticação de um TA.

(1) $\mathcal{P}$: $\quad \rightarrow \langle \mathbb{G} \ , \ g \ , \ r \rangle \quad ; \quad a \leftarrow \mathbb{Z}_r^* \quad ; \quad a \rightarrow \beta = g^a$

Assinatura O emissor $\mathcal{E}$ obtém uma assinatura σ para a mensagem m usando o segredo de $\mathcal{P}$

(1) $\mathcal{P}$: $\upsilon \leftarrow \mathbb{Z}_r^* \quad ; \quad \upsilon \rightarrow \gamma = g^{\upsilon}$
(2) $\mathcal{E}$: $w, u, c \leftarrow \mathbb{Z}_r^* \quad ; \quad \mu \leftarrow \gamma^w \, g^u \, \beta^c \quad ; \quad y \leftarrow H(\mu \| m) \quad ; \quad \rightarrow x = w^{-1} \, (y - c)$
(3) $\mathcal{P}$: $a, \upsilon \rightarrow s = \upsilon - a \, x$
(4) $\mathcal{E}$: $\gamma \stackrel{?}{=} g^s \cdot \beta^x \quad ; \quad z \leftarrow w \, s + u \quad ; \quad \mu, m \rightarrow \sigma = z \, \| \, y$

Verificação

(1) $\mathcal{V}$: $(z, y) \leftarrow \sigma \quad ; \quad \mu' \leftarrow g^z \cdot \beta^y \quad ; \quad y' \leftarrow H(\mu' \| m) \quad ; \quad y \stackrel{?}{=} y'$

Alguns pontos sobre os objectivos do protocolo através do grau de conhecimento dos agentes envolvidos e das suas crenças quanto à autenticidade da informação.

1. A produção da assinatura σ é um protocolo síncrono que envolve colaboração entre o emissor $\mathcal{E}$, que conhece a mensagem a assinar m, e o "prover" $\mathcal{P}$, que conhece a chave privada a usada na assinatura.

   O emissor usa o segredo do "prover" (a sua chave privada a) sem nunca o conhecer em qualquer fase do protocolo. Pelo seu lado o "prover" não deve conhecer m em qualquer fase do protocolo ou o seu "hash".

   Na verificação, o verificador $\mathcal{V}$ conhece a mensagem m; assume-se que a recebeu do emissor via um canal privado (por exemplo, usando uma cifra) e conhece a informação pública do "prover" (a sua chave pública).
2. O verificador tem de acreditar na autenticidade da chave oública do "prover". Por isso um TA tem de fornecer um certifificado que associe o "prover" à sua chave pública.

   O emissor tem de ter garantias que o "prover" agiu de boa fé antes de publicitar a assinatura da mensagem.

Por estas razões, o protocolo de assinatura inclui:

- Um acto de compromisso – passo (1) – onde o "prover" gera um segredo segredo υ (uma "chave de sessão") e compromete-se com a respectiva "chave pública" γ.
- A produção pelo emissor – passo (2) – de um "hash" da mensagem m que é cifrado; só o respectivo criptograma x é conhecido pelo "prover".

- A assinatura σ é construída em duas fases; na primeira – passo (3) – é executada pelo "prover"; a segunda – passo (4) – é executada pelo emissor.
  O "prover" produz, no passo (3), uma pré-assinatura s do "hash" cifrado x, usando o segredo υ, com que se comprometeu, e a sua chave privada a. O emissor, no passo (4), assegura-se da autenticidade de s usando o compromisso γ, antes de, com segredos próprios, usar s para construir a assinatura final.

Para analisarmos a **correcção** deste protocolo convém notar que:

1. Dada a forma como γ e β são gerados, verificam-se, no passo (2), as seguintes igualdades
$$\mu \;=\; g^{w\,\upsilon+u+a\,c} \qquad , \qquad w\,x+c \;=\; y$$
2. As igualdades que resultam dos passos (3) e (4) são, respectivamente,
$$a\,x+s \;=\; \upsilon \qquad , \qquad w\,s+u \;=\; z$$
3. No passo (4), a verificação da pré-assinatura calcula $g^s\,\beta^x = g^{s+a\,x}$ e compara-o com $\gamma = g^{\upsilon}$. Se o "prover" agiu de boa fé no passo (3), produz um s que satisfaz a igualdade $a\,x+s=\upsilon$; portanto esta "boa fé" consta-se na comparação $\gamma \;=?=\; g^s\,\beta^x$.
4. A verificação da assinatura σ calcula $\mu' = g^{z+a\,y}$. Atendendo às igualdades anteriores, tem-se
$$z+a\,y \;=\; w\,s+u+a\,w\,x+a\,c \;=\; w\,(s+a\,x)+u+a\,c \;=\; w\,\upsilon+u+a\,c$$

Portanto

$$\mu' = g^{z+a\,y} = g^{w\,v+u+a\,c} = \mu$$

Os valores y e y' são "hashs" de m calculados, respectivamente, com as chaves μ e μ'; dado que as chaves coincidem, os "hashs" também têm de coincidir.

Quanto à **segurança** convém referir a alguns pontos:

Poderia parecer, à primeira vista, que o segredo c é irrelevante. A correcção estaria assegurada se, no passo (2), c não fosse gerado aleatoriamente e se fixasse uma constante c usada em todas as instâncias do protocolo. Note-se porém que os valores de x e y são ambos públicos após a produção da assinatura; se c fosse conhecido, então seria trivial, a um atacante, calcular o segredo w, dada a relação $w\,x + c = y$. Em seguida, dado que s e z são públicos, o atacante pode calcular o outro segredo u, usando a relação $z = w\,s + u$, e finalmente μ.

Do mesmo modo, quaisquer circunstâncias que permitam descobrir um dos três segredos w, u, c, pemitem descobrir os dois restantes segredos e, por isso, permitem construir um ataque semelhante a este.

Conhecendo w, u, c, o atacante dispõe de informação que lhe permite gerar mensagens arbitrárias $m_1 \neq m$, o respectivo "hash" cifrado y_1 e a assinatura $z_1 \| y_1$, sem passar pela intervenção do "prover". Para isso, basta escolher um w_1 que verifique $w_1\,x + c = y_1$ e calcular $z_1 = w_1\,s + u$. Como o valor de $s + a\,x$ se mantém inalterado, o algorimo de verificação continua a ter sucesso com esta nova assinatura e com a mensagem m_1.

5.9 Protocolos de Acordo de Chaves

O protocolo Diffie-Hellman (protocolo 6, página 303) foi a primeira técnica criptográfica de chave pública publicada. Foi introduzido em 1976 e está na base do interesse em grupos cíclicos como suporte à especificação de técnicas de chave pública.

O protocolo Diffie-Hellman pertence à classe dos "protocolos de acordo de chaves" que, em linhas gerais, são

125 Noção

Um **protocolo de acordo de chaves** *é um protocolo síncrono envolvendo dois agentes legítimos (designados por* **principais***) que, através de uma troca sucessiva de mensagens usando um canal público num meio hostil, obtêm informação suficiente para lhes permitir calcular um segredo comum* λ.

Esta definição genérica tem de ser concretizada numa série de condições de correcção e segurança. As condições de correcção determinam como se devem comportar os agentes numa execução legítima do protocolo. As condições de segurança estipulam propriedades que devem ser verificadas em qualquer execução do protocolo (legítima ou não).

Para as definir é necessário concretizar um pouco mais o que significa alguns conceitos que acabámos de referir: agente, canal público, mensagem, meio hostil, segredo, etc.

Para isso começamos por concretizar um pouco mais o que é um *"protocolo"*.

Cada execução específica do protocolo designa-se por **instância**. Cada instância é uma sequência finita de **passos**; cada passo de protocolo tem um *autor* (identificado pelo seu nome) e é uma computação que produz uma *mensagem*.

Cada protocolo especifica um conjunto finito de computações disponíveis para a criação de mensagens e disponibiliza um determinado número de constantes designadas por "parâmetros de execução". Cada mensagem é criada, com estas computações, a partir dos parâmetros de execução, do nome do autor, da sua informação privada, de mensagens anteriores e de consultas a oráculos.

Os oráculos disponíveis são específicos de cada protocolo mas, tipicamente, incluem:

- Um gerador $\mathfrak{N}$ que, em cada invocação, gera um novo inteiro. Cada "output" de $\mathfrak{N}$ designa-se por **nounce**[40]. A sequência de "nounces" é suficientemente aleatória e sem repetições.
- Um "trusted agent" `TA` que, ao receber um nome A, determina uma chave pública de A.

O facto da computação executado no passo p usar uma mensagem calculada no passo q, estabelece naturalmente uma **relação de precedência** entre estes passos: p é **anterior** ou **precede** q. Uma **história** do protocolo é uma sequência (eventualmente infinita) de passos de protocolo, resultantes de uma sequência de instâncias, mantendo a relação de precedência específica de cada instância.

[40] Number Only Used onCE.

Nota

Ou seja: se numa instância particular, o passo p precede o passo q, então na história p ocorre antes de q. Isto não impede que ocorram, entre p e q, passos provenientes de outras instâncias do protocolo envolvendo, eventualmente, outros agentes.

Um **canal público** é uma memória de longo prazo onde são depositadas as diferentes mensagens de uma história do protocolo conjuntamente com informação sobre a relação de precedência entre as mensagens. A menos que tal informação conste na mensagem, o canal público não conhece o seu autor.

Cada agente é determinado pelo seu **nome**, pela sua **informação privada** e pelos seus **privilégios** de acesso ao canal público. Estes privilégios podem ser:

- **Escuta:** o agente tem conhecimento de todas ou parte das mensagens no canal público. Este privilégio pode ser *local*, se se refere apenas à instância presente do protocolo, ou *global*, se se refere a todas as instâncias presente e passadas do protocolo.
- **Execução:** o agente tem possibilidade de executar novos passos de protocolo ou repetir passos anteriores.
- **Controlo:** o agente tem a capacidade de negar, selectiva ou globalmente, privilégios de outros agentes.

Nos protocolos de acordo de chave entende-se que os **agentes principais** são caracterizados pelo seu nome, pela sua informação privada e têm privilégio de escuta local e execução de novos passos do protocolo. O *"meio hostil"* é personificado num agente designado por **atacante** que tem privilégios global de escuta, execução e tem privilégio controlo selectivo de escuta e execução de um determinado conjunto de agentes.

□

Para caracterizar, dentro da noção genérica de protocolo, o que é um protocolo de acordo de chaves começamos por definir uma condição que especifica o seu objectivo.

Correcção

O segredo é comum no sentido em que, em cada instância do protocolo, o mesmo valor de λ é determinado por ambos os agentes principais.

As condições de segurança vão ser expressas em termos de uma história do protocolo envolvendo um único atacante C e, em cada instância, dois agentes principais A e B.

Confidencialidade

Em nenhum momento, C conhece a chave λ acordado pelo par de agentes A, B.

Autenticidade dos agentes

Em cada instância, o agente A tem prova que o outro interveniente no protocolo é B e, vice-versa, B tem prova que o outro interveniente é A.

Autenticidade do segredo

Cada um dos agentes (A ou B) tem prova que o outro calculou o mesmo segredo λ que ele próprio calculou.

O protocolo determina uma **chave estática** λ quando, para o mesmo par de agentes A e B, toda a instância do protocolo envolvendo estes agentes determinam a mesma chave. O segredo é uma **chave efémera** (ou "de sessão") quando, para o mesmo par de agentes, cada instância do protocolo determina uma chave uniformemente distribuída dentro de um domínio vasto de possibilidades. Isto é, a chave é efémera quando a sua incerteza é igual (ou ligeiramente inferior) ao seu comprimento e é estática se a sua incerteza for nula.

A efemeridade da chave está ligada à sua confidencialidade. Uma chave estática pode sempre, por motivos alheios à execução do protocolo, passar a ser conhecida pelo atacante C. Por exemplo, um dos agentes pode ser indiscreto no seu uso. Uma chave efémera é menos sensível a essas "fugas de informação" no sentido em que o seu eventual conhecimento pelo atacante deixa de ser relevantes para utilizações futuras. A efemeridade da chave levanta também a questão da autenticidade temporal; isto é, a autenticidade relativa às instâncias particulares do protocolo.

Surgem assim variantes temporais das condições de segurança anteriores.

Novidade
: Numa determinada instância do protocolo, A e B têm a certeza que λ não pode ser determinada a partir de instâncias anteriores do mesmo protocolo.

Autenticidade temporal dos agentes
: Numa determinada instância do protocolo, A tem a certeza que o outro agente é B participando na mesma instância do protocolo. Analogamente, para B.

Autenticidade temporal do segredo

Numa determinada instância do protocolo, cada agente tem prova que o outro calculou, na mesma instância, o mesmo segredo que ele próprio determinou.

A indiscrição coloca-se também em relação à informação privada dos agentes principais. Isto levanta a questão da manutenção da confidencialidade dos segredos acordados com informação privada. Temos assim uma condição de segurança adicional.

Confidencialidade a longo prazo ("foward secrecy")

O conhecimento, pelo atacante, da chaves privada de um conjunto de agentes não lhe permite ter conhecimento das chaves de sessão acordadas em instâncias anteriores onde algum desses agentes tenha intervido.

□

Muitos protocolos de acordo de chaves não necessitam de uma concretização específica das estruturas matemáticas subjacentes; são definidos, de forma abstracta, recorrendo a cifras, funções de "hash" e assinaturas com segurança perfeita. Nomeadamente, muitos protocolos importantes usam exclusivamente técnicas simétricas (cifras e funções de "hash").

Porém estes protocolos exigem um mecanismo prévio de distribuição das chaves simétricas o que conduz, naturalmente, às questões de autenticidade desse mecanismo.

Nesta secção vamos considerar protocolos baseados em técnicas de chave pública e, em particular, protocolos assentes em grupos cíclicos. Vamos começar por considerar o protocolo de Diffie-Hellman, e algumas variantes simples, e tentar verificar quais destas condições são satisfeitas e quais são violadas. Vamos assumir um grupo Diffie-Hellman $\langle \mathbb{G}, g, r \rangle$ de ordem prima; vamos assumir os elementos comuns referidos na página 309. Vamos, finalmente, assumir que todos os passos de protocolo lidam com valores apenas em dois domínios: $\mathbb{Z}_r$ ou numa órbita G de g de ordem r. Estas condições garantem que os protocolos não são vulneráveis a ataques baseados na pequena ordem dos grupos cíclicos.

Protocolo 18 : Diffie Hellman - chave estática

Setup Criação de pares de chaves de longa duração e iniciação do TA com a respectiva chave pública.

(1) $A: a \leftarrow \mathbb{Z}_r^* \quad ; \quad \rightarrow \beta = g^a \quad ; \quad B: b \leftarrow \mathbb{Z}_r^* \quad ; \quad \rightarrow \gamma = g^b$
(2) O TA reconhece e autentica os pares (A, β) e (B, γ)

Run

(1) $A: \mathsf{TA}(B, \gamma) =?= 1 \quad ; \quad \lambda_a \leftarrow \gamma^a$
(2) $B: \mathsf{TA}(A, \beta) =?= 1 \quad ; \quad \lambda_b \leftarrow \beta^b$

Neste, e nos restantes protocolos desta secção, amos assumir também que existem oráculos

- $\mathfrak{N}$: produz um "nounce" em cada activação.
- TA é uma decisão que sob "input" do nome A e de uma chave pública β decide se o par (A, β) é autêntico.

O TA determina também, na fase de "setup", o grupo Diffie-Hellman $\langle \mathbb{G}, g, r \rangle$ sob o qual o protocolo se desenvolve.

O protocolo estático acaba por não produzir qualquer mensagem para o canal público. Cada um dos agentes obtém, do TA, a chave pública do outro e calcula imediatamente o segredo com a sua chave privada.

A propriedade da correcção é trivialmente satisfeita já que $\lambda_a = \gamma^a = (g^b)^a = (g^a)^b = \beta^b = \lambda_b$. Como não existem mensagens, a única informação disponível ao intruso C é a que provém do TA; C pode obter também as chaves públicas; no entanto só conhecerá o segredo se conseguir resolver o problema CDHP $g, g^a, g^b \rightarrow g^{a\,b}$; portanto a condição de confidencialidade é garantida.

A autenticidade dos agentes é garantida porque as chaves públicas são autenticadas pelo TA. Não há garantias da autenticidade do segredo já que nenhum dos agentes pode ter garantia de que o outro o chega a calcular.

No entanto, a desvantagem essencial deste protocolo é o facto de a chave acordade ser estática: entre os mesmos dois agentes o segredo acordado é sempre o mesmo. Nomeadamente qualquer indiscrição na informação privada, compromete todas os usos anteriores da chave; a longo prazo, o protocolo é inseguro.

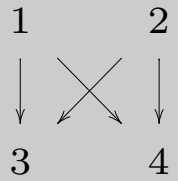
Para suprir esta falha de segurança, a sugestão imediata será o de substituir as chaves de longa duração a e b, por chaves locais a cada instância do protocolo. Como consequência imediata, estas chaves não podem ser autenticadas.

Protocolo 19 : Diffie Hellman - versão efémera

Setup O TA escolhe um grupo Diffie-Hellmam $\langle \mathbb{G}, g, r \rangle$ e torna pública esta informação.

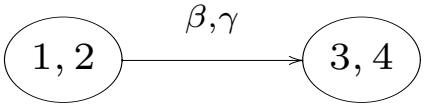
Run

$$
\begin{array}{llll}
(1) & A: & a \leftarrow \mathbb{Z}_r^* \;\; ; & a \rightarrow \beta = g^a \\
(2) & B: & b \leftarrow \mathbb{Z}_r^* \;\; ; & b \rightarrow \gamma = g^b \\
(3) & A: & \lambda_a \leftarrow \gamma^a & \\
(4) & B: & \lambda_b \leftarrow \beta^b &
\end{array}
$$



Em primeiro lugar note-se que, ao contrário do protocolo estático, existe uma sequência de passos e uma clara relação de precedência entre eles. De facto a precedência é definida $(1), (2) \rightarrow (3)$ e $(1), (2) \rightarrow (4)$.

Como não está estabelecida nenhuma precedência entre (1) e (2) nem entre (3) e (4), estes pares de passos podem-se considerar "simultâneos" em termos de uma execução legítima do protocolo. Assim a execução pode ser representada por



Este protocolo permite, porém, execuções ilegítimas que designaremos por **ataques**. Para analisar como se desenvolvem estes ataques vamos considerar uma notação genérica que nos permite uma representação mais detalhada da sucessão de passos do protocolo.

Cada agente X (A, B ou um atacante) gera ou calcula segredos s durante os vários passos do protocolo. Representamos por $X.s$ o valor do segredo s que X conhece. Caso o agente execute o mesmo passo vários vezes, os valores do mesmo segredo s gerado em sucessivos passos é denotado por $X.s_1, X.s_2, ,$ etc.

Vamos ter também uma definição genérica dos passos deste protocolo. Note-se que o protocolo tem dois tipos de passos que designaremos por T e S e que são instanciados com os agentes principais ao passo e com a informação que vão buscar ao canal público.

$$T(X) \doteq \left\{X:\ k \leftarrow \mathbb{Z}_r^*\ ;\ k \rightarrow g^k\right\} \quad , \quad S(X,u) \doteq \left\{X:\ \lambda \leftarrow u^{X.k}\right\}$$

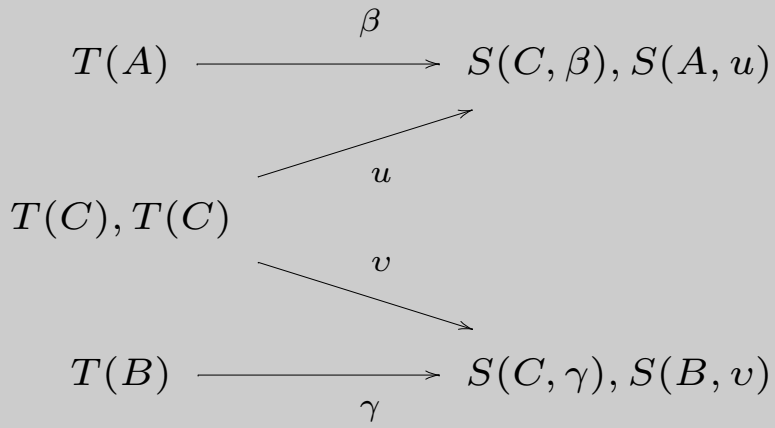
Com esta notação os passos (1) e (2) são, respectivamente, $T(A)$ e $T(B)$, identificando as chaves privadas efémeras a e b com k; isto é, $a = A.k$ e $b = B.k$. Do mesmo modo os passos (3) e (4) são, respectivamente, $S(A,\gamma)$ e $S(B,\beta)$, identificando $\lambda_a = A.\lambda$ e $\lambda_b = B.\lambda$.

Usaremos a notação $T(X) \xrightarrow{u} S(Y,u)$ para indicar que u foi publicidado por $T(X)$ e é usado em $S(Y,u)$. As relações de precedência são, então, para todo X, Y, u

$$T(X) \xrightarrow{u} S(Y,u) \quad , \quad T(X) \longrightarrow S(X,u)$$

Vamos considerar, agora, um atacante C que inicia dois passos T. É agora possível uma história como a que representa-mos no seguinte diagrama[41].

Ataque 1 Homem-No-Meio



Existe uma ocorrência do passo T e do passo S para A e para B e duas ocorrências, de ambos os passos, para C. Os passos T produzem

$$\beta = g^{A.k} \qquad u = g^{C.k_1} \qquad \upsilon = g^{C.k_2} \qquad \gamma = g^{B.k}$$

[41] Por simplicidade não se representa explicitamente as dependências entre um passo T e um passo S envolvendo o mesmo agente.

Os passos S produzem

$$A.\lambda = u^{A.k} = g^{C.k_1\ A.k} \qquad C.\lambda_1 = g^{A.k\ C.k_1} \qquad C.\lambda_2 = g^{B.k\ C.k_2} \qquad B.\lambda = g^{C.k_2\ B.k}$$

Por isso, o par de passos $S(A,u), S(C,\beta)$ acordou o mesmo segredo $g^{C.k_1\ A.k}$ enquanto que o par de passos $S(B,\upsilon), S(C,\gamma)$ acordou o segredo $g^{C.k_2\ B.k}$.

Desta forma o atacante C foi capaz de acordar segredos tanto com A como com B. Como não existe qualquer autenticação dos agentes, nenhum dos agentes proncipais tem capacidade para identificar o seu parceiro no acordo. Portanto, A e B podem pensar que estão a comunicar entre si mas realmente estão a comunicar com C.

A não-autenticidade dos agentes traduz-se aqui em falha na autenticidade do segredo; de facto, nem A nem B podem confirmar que o segredo usado pelo outro é o mesmo que eles próprios calcularam e, como se verifica, o segredo é diferente.

□

Comparando as duas versões do protocolo Diffie-Hellam vê-se que o protocolo 18 fornece autenticação dos agentes, mas não garante autenticação do segredo nem novidade nesse segredo; em contra-partida o protocolo 19 produz uma chave fresca em cada instanciação mas não garante a autenticidade dessa chave nem dos agentes.

Para tentar aliar as vantagens de ambas as versões (e eliminar as falhas de segurança) vamos considerar um protocolo

que contém, para além do grupo Diffie-Hellman, um esquema de assinaturas e uma cifra. Usando a notação usual da escrita de protocolos usaremos a seguinte convenção

$\nu\, x \cdot m(x)$	a mensagem que resulta de instanciar $m(x)$ com um nounce x
$\{m\}_k$	criptograma resultante da cifra de m com a chave k
$\{x\}_k^{-1}$	texto claro obtido do criptograma x com a chave k
$[m]$	"hash" da mensagem m
$[m]_k$	MAC ("message authentication code") da mensagem m com a chave k
$(m)_X$	assinatura da mensagem m pelo agente X

Vamos também assumir que existe um oráculo Ver que verifica assinaturas. Este oráculo é implementado como uma decisão $\mathrm{Ver}(X, m, s)$ que. sob "input" de um nome X, uma mensagem m e de uma assinatura s, decide se s é uma assinatura autentica de m por X.

O protocolo *Station-To-Station* (STS) é um protocolo baseado no protocolo Diffie-Hellman mas onde os agente principais executam "passos complementares". O protocolo é sequencial no sentido em que existe um agente específico que o inicia e os passos seguintes são sempre respostas ao passo que o precede; existe só uma história que define uma execução legítima.

Protocolo 20 : Station-To-Station - versão anónima

(1) A: $a \leftarrow \mathbb{Z}_r^*$; $a \rightarrow \beta = g^a$

(2) B: $b \leftarrow \mathbb{Z}_r^*$; $k \leftarrow \beta^b$; $\rightarrow \sigma = g^b \,\|\, \left\{ \left(g^b \| \beta \right)_B \right\}_k$

(3) A: $(\gamma, t) \leftarrow \sigma$; $k \leftarrow \gamma^a$; $\mathsf{Ver}(B, \gamma \| \beta, \{t\}_k^{-1}) \stackrel{?}{=} 1$; $\rightarrow \upsilon = \left\{ (g^a \| \gamma)_A \right\}_k$

(4) B: $\mathsf{Ver}(A, \beta \| g^b, \{\upsilon\}_k^{-1}) \stackrel{?}{=} 1$

Convencionalmente os protocolos de acordo de chave representam-se não nesta forma extensa mas numa forma mais abstracta onde são apresentadas apenas as mensagens que são publicadas no canal público.

Protocolo 21 : STS - versão anónima

(1) A: $\nu\, a \cdot \beta$ sendo $\beta = g^a$

(2) B: $\nu\, b \cdot \gamma \| \{(\gamma \| \beta)_B\}_{k_B}$ sendo $\gamma = g^b$ e $k_B = \beta^b$

(3) A: $\{(\beta \| \gamma)_A\}_{k_A}$ sendo $k_A = \gamma^a$

(4) B: ε

Uma diferença importante entre estas duas representações é o facto de toda a informação privada ser, agora, gerada sob a forma de "nounces" e não "simplesmente" aleatória (o que permite repetições). Nesta representação são

consideradas implícitas todas as decomposições de mensagens em componentes e todas as verificações de assinaturas. Por isso, no último passo, está implícita a verificação da assinatura produzida no passo anterior; isto é, o passo existe só que, mesmo que a verificação tenha sucesso, não produz qualquer mensagem.

Basicamente STS é o protocolo Diffi-Hellman aumentado com autenticação dos agentes e do segredo. A correcção está assegurada pela equação $(g^a)^b = (g^b)^a$. A novidade da chave é assegurada pela geração dos "nounces" nos passos (1) e (2). A autenticidade dos agentes é reconhecida porque ambos assinam uma mensagem previamente conhecida por ambos ($g^b \| g^a$ ou $g^a \| g^b$) e essa assinatura é verificada pelo outro agente no passo seguinte. A confirmação da chave concretiza-se porque estas assinaturas são cifradas com a chave acordada; para verificar a assinatura é preciso decifrá-la primeiro.

A presença da cifra nos passos (2) e (3) é essencial; sem ela, no passo (3), um atacante "homem-no-meio" poderia substituir a assinatura $(\gamma \| \beta)_A$ pela sua própria assinatura. Deste modo A e B completam o protocolo, mas A acredita que comunicou com B enquanto que B acredita que comunicou com C.

Este tipo de falha permanece mesmo no protocolo original. Considere-se, por exemplo, o ponto de vista do agente A; a menos que ele saiba previamente qual é o nome do agente com quem deve acordar o protocolo, não lhe é possível no passo (3) verificar a assinatura produzida no passo (2). O agente A não garante a autenticidade de "agentes imprevistos".

Isto ocorre porque a informação que cada agente tem sobre o outro interveniente tem de ser fornecida por um outro canal externo ao do protocolo. Idealmente a identificação dos intervenientes no protocolo deveria estar contida nas

próprias mensagens do protocolo.

Pode-se pensar numa modificação ao protocolo onde cada mensagem contenha explicitamente, no cabeçalho, a origem e o destino previstos. Uma primeira abordagem seria,

(1)	$A:$	$\nu\, a \cdot A\|\|B\|\|\beta$	sendo $\beta = g^a$
(2)	$B:$	$\nu\, b \cdot B\|\|A\|\|\gamma\|\| \{(\gamma\|\|\beta)_B\}_{k_B}$	sendo $k_B = \beta^b$ e $\gamma = g^b$
(3)	$A:$	$A\|\|B\|\| \{(\beta\|\|\gamma)_A\}_{k_A}$	sendo $k_A = \gamma^a$

Este protocolo é obviamente inseguro porque os nomes não estão autenticados; assim qualquer intruso pode re-enviar mensagems substituindo o nome do agente origem, do agente destino ou de ambos. O melhor que se pode dizer é que este protocolo não é mais inseguro que o STS original mas não resolve o problema da identificação dos agentes.

Obviamente que se podia incluir os cabeçalhos, $A||B$ ou $B||A$, na componente assinada e cifrada. De facto, a autenticação do nome dos agentes suprime também, e em parte, a necessidade de cifrar a assinatura para confirmar a autenticidade do segredo; isto porque que cada um dos agentes sabe que o outro conhece exactamente os mesmos valores de β e γ que ele próprio conhece.

Isto não é exactamente a mesma coisa do que a autenticidade da chave acordada. O que cada agente sabe é que o

outro tem os elementos necessários para calcular a chave certa; não sabe, porém, se o outro (agindo de má fé) não usa outra chave. Para obter a confirmação das chaves é necessário juntar um MAC da informação assinada.

Cada agente publicita duas mensagens; não é necessário que em ambas seja identificado o emissor e o destinatário. A versão simplificada de um protocolo STS com identificadores e autenticação MAC é

Protocolo 22 : STS - versão com identificadores

			sendo
(1)	A:	$\nu\, a \cdot A \parallel \beta_A$	$\beta_A = g^a$
(2)	B:	$\nu\, b \cdot B \parallel \beta_B \parallel (m_B)_B \parallel [m_B]_{k_B}$	$\beta_B = g^b$, $m_B = \beta_B \Vert \beta_A \Vert A$ e $k_B = (\beta_A)^b$
(3)	A:	$(m_A)_A \parallel [m_A]_{k_A}$	$m_A = \beta_A \Vert \beta_B \Vert B$ e $k_A = (\beta_B)^a$
(4)	B:	ε	

Este protocolo garante "foward secrecy"; de facto as únicas chaves privadas são as usadas para gerar assinaturas e a chave de sessão $g^{a\,b}$. A chave de sessão $g^{a\,b}$ depende apenas dos "nounces" gerados nessa instância do protocolo; não depende das chaves usadas para as assinaturas e, se o gerador de "nounces" for suficientemente aleatório, não depende de "nounces" noutras instâncias do protocolo.

□

O protocolo STS usa um esquema de assinaturas genérico. Se essa assinatura fosse, por exemplo, o DSA existe

uma óbvia redundância já que muitas das primitivas criptográficas do protocolo Diffie-Hellman básico se repetem na assinatura DSA. Assim, na perspectiva de melhorar a eficiência do STS, parece razoável procurar harmonizar as primitivas usadas no protocolo de acordo de chaves básico, com as que são usadas nas assinaturas digitais.

Uma primeira tentativa de harmonização é o protocolo Arazi, integrado no DSS (Digital Signature Standard).

Protocolo 23 : Arazi

Setup TA escolhe um grupo de Diffie-Hellman $\langle \mathbb{G}, g, r \rangle$ de ordem prima. A e B geram chaves privadas de longa duração a e b, publicitam as respectivas chaves públicas $\beta_A = g^a$ e $\beta_B = g^b$. Os pares (A, β_A) e (B, β_B) são autenticadas pelo TA. Assume-se os elementos comuns referidos na página 309.

Run

(1) $A: \ \nu\, x \cdot \lambda_A \| s_A$ sendo $\lambda_A = g^x$ e $x\, s_A = H(\lambda_A) + a\, f(\lambda_A)$

(2) $B: \ \nu\, y \cdot \lambda_B \| s_B$ sendo $\lambda_B = g^y$ e $y\, s_B = H(\lambda_B) + b\, f(\lambda_B)$

(3) $A: \ \varepsilon$

A verificação das assinaturas s_X para a mensagem λ_X (sendo X o agente A ou o agente B) é $\lambda_X^{s_X} =?= h(\lambda_X)\, \beta_X^{f(\lambda_X)}$.

A chave acordada é a do protocolo Diffie-Hellman: $k_A = \lambda_B^x$ e $k_B = \lambda_A^y$.

Este protocolo distingue-se do protocolo Diffie-Hellman básico apenas pelo facto de juntar às mensagens g^x e g^y as suas assinaturas. Se os agentes principais forem conhecidos *à priori*, então o protocolo garante a autenticidade dos agentes.

O protocolo falha em termos de "foward secrecy". Note-se que se uma das chapas privadas de longa duração (por exemplo, a) for comprometida no futuro então a chave acordada k_A está comprometida. Isso deriva da equação $x\, s_A = H(\lambda_A) + a\, f(\lambda_A)$ usada para calcular a assinatura s_A: se a for conhecido então x é conhecida já que todos os restantes elementos são públicos. Conhecido x, calcula-se $k_A = \lambda_B^x$.

As equações de geração de assinatura fornecem a redundância que permite este tipo de ataques. Pode-se provar, por exemplo, que se, numa instância particular, a chave acordada $k_A = k_B$ for comprometida, então estas equações podem fornecer informação que permita calcular esta mesma chave em instâncias futuras.

□

Se comparar-mos a assinatura aqui calculada com o DSA (protocolo 10, pag. 316) vemos que a assinatura do protocolo de Arazi não gera randomização; limita-se a usar o mesmo "nounce" (x ou y) que é usado na geração da chave acordada. Pode-se tentar resolver esta questão com dois "nounces" distintos em cada agente: um para a assinatura e outro para a chave.

Neste caso, a aleatorização da assinatura (com os "nounces" v e u) evita que os "nounces" da chave (x e y) sejam determinados mesmo quando ambas as chaves privadas a e b estão comprometidas. Portanto este protocolo resolve

as falhas na "foward secrecy" do protocolo de Arazi.

Protocolo 24 : Hirose-Yoshida

Setup coincide com a fase "setup" no protocolo de Arazi (protocolo 23)

Run

			sendo
(1)	A:	$\nu\, x \cdot \lambda_A \| A$	$\lambda_A = g^x$
(2)	B:	$\nu\, y, \upsilon \cdot \lambda_B \| \sigma_B \| s_B \| B$	$\lambda_B = g^y$, $\sigma_B = H(g^\upsilon \| \lambda_B \| \lambda_A)$
			$s_B = \upsilon - \sigma_B\, y - \sigma_B^2\, b$
(3)	A:	$\nu\, u \cdot \sigma_A \| s_A$	$\sigma_A = H(g^u \| \lambda_A \| \lambda_B)$, $s_A = u - \sigma_A\, x - \sigma_A^2\, a$
(4)	B:	ε	

A verificação das assinatura de B calcula g^υ como $g^{s_B}\,(\lambda_B\,\beta_B^{\sigma_B})^{\sigma_B}$ e, depois, compara σ_B com $H(g^\upsilon \| \lambda_B \| \lambda_A)$. Para a assinatura de A procede-se de forma análoga.

A chave acordada é a do protocolo Diffie-Hellman: $k_A = \lambda_B^x$ e $k_B = \lambda_A^y$.

□

Para finalizar esta breve introdução ao protocolos de acordo de chaves é importante referir que as técnicas aqui

propostas são apenas uma das componentes dos protocolos operacionais. Algumas das razões

(i) Dado que todos os protocolos de acordo de chaves são parametrizados por um conjunto de técnicas (cifras, funções de "hash", assinaturas) e envolvem escolhas sobre parâmetros das estruturas matemáticas, os protocolos operacionais têm de envolver trocas de mensagens que conduzam a um acordo sobre as técnicas criptográficas usadas e sobre os valores dos parâmetros.

Estes "acordos de parâmetros" são sujeitos a ataques análogos aos dos protocolo principal. Por isso é necessário incluir aqui, igualmente, componentes dirigidas à autenticidade.

(ii) Os protocolos operacionais têm de responder a um tipo de ataques que não está normalmente presente nos protocolos que aqui estudámos. Nomeadamente os protocolos operacionais, como o nome indica, têm de responder a questões ligadas à operacionalidade; nomeadamente, se é vulnerável a ataques de "denial of service" (DoS).

É razoável assumir que não existe nenhuma defesa eficaz contra os ataques DoS se assumir-mos que o atacante tem controlo completo do canal público. No entanto vários mecanismos têm sido propostos para, pelo menos, minimizar as probabilidade de um ataque com sucesso.

Na perspectiva de um ataque DoS, os agentes principais são vistos de forma assimétrica: um deles é um **"servidor"** S, que é objecto do ataque, e o outro é um **"cliente"** C que age de forma hostil tentando esgotar os recursos do servidor.

Algumas abordagens possíveis são:

1. *stateless connections*
   O cliente armazena toda a informação de estado (mensagens anteriores e "nounces") que o servidor necessita e envia-as para o servidor conforme é necessário. Isto requer que a informação esteja cifrada e que sejam necessários mecanismos de autenticação que garantam ao servidor que está a receber a informação certa. No entanto o servidor não necessita de armazenar localmente estado e poupa nos seus recursos. Em contrapartida há custos adicionais quer em esforço computacional como em tráfego que façam com que, de facto, os recursos do servidor sejam esgotados ainda mais depressa.
2. *cookies*
   O servidor atrasa a necessidade de estado local enviando um "cookie" ao cliente, sempre que ele tenta estabelecer uma ligação. Esse *cookie* contém informação identificadora da ligação e um "semi-nounce" cifrados com uma chave privada do servidor. Nesta fase S ainda não se comprometeu com o protocolo nem criou informação de estado. O cliente tem de devolver o "cookie" na próxima mensagem e dentro de um intervalo de tempo limitado. O "semi-nounce" não necessita de ser único em cada "cookie" mas é actualizado muito frequentemente.
3. *outras autenticações*
   Os *cookies* são uma forma básica de autenticação prévia. Pode-se estender outras formas de autenticação a todas as mensagens do protocolo. Isto é, obviamente, muito custoso mas tem a vantagem de o servidor poder parar imediatamente o protocolo logo que detecta uma mensagem não autenticada. Em alternativa o servidor pode, de forma incremental, ir aumentando as suas exigências de autenticação consoante a sua carga de ligações activas aumenta. Por exemplo, pode requerer autenticação via dos protocolos de identificação desafio-resposta.

5.10 Protocolos para Trocas Cifradas de Chaves

O objectivo geral dos protocolos *"Encrypted Key Exchange"* (EKE) é o de um acordo ou transporte de uma chave de sessão usando, como forma de autenticação das mensagens, uma "password" partilhada pelos agentes principais.

O argumento por detrás desta opção, por oposição aos protocolos de acordo de chaves descritos na secção anterior, vai no sentido de considerar que os mecanismos de autenticação aí usados, baseados em assinaturas digitais, requererem chaves privadas de longa duração com um número elevado de bits e envolvem mecanismos complexos de gestão e certificação dessas chaves.

A alternativa aqui proposta usa "passwords", com pouca incerteza[42], para autenticação. O facto de a "password" ser pequena facilita a sua gestão pelos agentes principais; nomeadamente, a "password" é fácil de recordar e, normalmente, não requer mecanismos de protecção muito sofisticados.

As assumpções básicas, que condicionam a segurança desta família de técnicas, são

1. A "password" (ou uma sua imagem) é conhecida só pelos agentes principais. Um adversário que, de qualquer forma, conheça "password" pode intervir no protocolo fazendo-se passar por um dos agentes principais.
2. A "password" é usada como chave de uma cifra simétrica; devido à reduzida dimensão do espaço de chaves, a cifra é vulnerável a um ataque por força-bruta com texto conhecido.

[42] Obviamente, se já existisse uma chave partilhada π de elevada incerteza, o protocolo seria irrelevante!

3. A "password" não é vulnerável a ataques de dicionário por adversários passivos. Isto é, um adversário que escute um número N de criptogramas gerados com a chave π, mantém uma incerteza sobre a chave $\vartheta(\pi)$ que é essencialmente constante, independentemente de N.

Quase sempre uma chave π, num espaço de chaves K, é usada para cifrar elementos de uma órbita $G = [g]$ de um grupo de Diffie-Hellman $\langle \mathbf{G}, g, r \rangle$. Assume-se que G está contido no domínio de todas as possíveis chaves $\pi \in K$, mas pode ser apenas uma pequena parte desse domínio.

Dado um criptograma $y = \pi\tilde{}(u)$, com $u \in G$, um atacante passivo, mesmo não conhecendo u, pode tentar encontrar possíveis chaves π para as quais $\pi^{-1}(y) \notin G$. Tais chaves, se existirem, serão imediatamente eliminadas do espaço de chaves K diminuindo a incerteza em π.

Por isso, a não vulnerabilidade a ataques de dicionário, implica a seguinte condição de segurança

$$\forall \pi, \pi\tilde{} \in K\ ,\ \forall u \in G \quad . \quad \pi^{-1}(\pi\tilde{}(u)) \subseteq G \tag{117}$$

□

Um adversário activo pode sempre reduzir ligeiramente a incerteza em π, gerando uma tentativa $\pi\tilde{}$, entrando no protocolo com esta este valor fazendo-se passar por um legítimo titular de π e, consoante o protocolo corre com

sucesso ou não, decidir se a sua tentativa é a "password" correcta. Desta forma, fazendo $2^{\vartheta(\pi)-1}$ tentativas, pode subir a propbabilidade de escolha da chave correcta acima de $1/2$.

Atendendo à nossa primeira assunção, isto implica poder completar o protocolo com sucesso fazendo-se passar por um dos titulares legítimos de π. Desta forma faz sentido considerar a seguinte condição de segurança para esta classe de protocolos:

O protocolo EKE diz-se **seguro contra um ataque de dicionário activo** se, iniciando um número de instâncias do protocolo que seja substancialmente inferior a $2^{\vartheta(\pi)-1}$, nenhum adversário consegue obter probabilidade superior a $1/2$ de terminar com sucesso.

Nos protocolos desta secção assume-se

Elementos Comuns

(1) Um grupo Diffie-Hellman $\langle \mathbb{G}, g, r \rangle$ de ordem r prima; $G = [g]$ denota a órbita do gerador g.
(2) Uma cifra $\{\cdot\}$, com um espaço de chaves K.
(3) Funções de *hash* $H : \mathbb{N} \longrightarrow G$ e $h : G \longrightarrow K$.
(4) Funções de *hash* $H_i : \mathbb{N} \longrightarrow \mathbb{B}^t$, $i = 1, 2, 3$, independentes.

As funções de *hash* $H_i : \mathbb{N} \longrightarrow G$ são independentes quando, para todo $i \neq j$, é computacionalmente intratável encontrar strings x, y tais que $H_i(x) = H_j(y)$.

A partir de uma única função de *hash*, $\theta : \mathbb{N} \longrightarrow \mathbb{B}^t$, pode-se sempre gerar uma sequência de funções de *hash* independentes. Por exemplo, pode-se definir $H_i(x) = \theta(i \| x)$ ou $H_i(x) = \theta(\{x\}_{k_i})$, sendo $k_i = h(H(i))$.

□

O paradigma de protocolo EKE é o chamado **protocolo de Bellovin & Merrit** e que é, como muitos outros, baseado no protocolo de Diffie-Hellman. Usa uma "password" π e uma cifra simétrica como única forma de autenticação.

Como o protocolo não usa esquemas de assinaturas, não requer chaves privadas e públicas de longa duração. Assim, todas as chaves privadas a, b são "nounces" e todas as chaves públicas (as mensagens g^a e g^b) são efémeras.

Essencialmente o protocolo cifra com π todas estas chaves públicas efémeras antes de as enviar pelo canal público. A sua autenticidade depende crucialmente da assunção de que π é só conhecido por A e B.

A autenticidade da chave Diffie-Hellman acordada $g^{a\,b}$ é verificada usando dois novos "nounces" $n, n^{\sim}$ que são trocados cifrados com uma chave derivada da chave Diffie-Hellman.

Protocolo 25 : EKE ("encrypted key exchange") Bellovin & Merrit

Seup TA escolhe uma *password* $\pi \in K$ e distribui-a, de forma confidencial, pelos agentes principais.

Run

(1) $A:\ \nu\, a \cdot A \parallel \{g^a\}_\pi$

(2) $B:\ \nu\, b, n \cdot \left\{g^b\right\}_\pi \parallel \{n\}_k$ sendo $k = h(g^{a\,b})$

(3) $A:\ \nu\, n^{\sim} \cdot \{n^{\sim} \| n\}_k$

(4) $B:\ \{n^{\sim}\}_k$

Uma variante deste protocolo força uma escolha particular de cifra $\{\cdot\}$ e da sua função de derivação de chaves h.

Protocolo 26 : PAK ("passowrd authenticated key") Boyco *et al.*

Setup TA escolhe π e distribui esta chave pelos agentes principais

Run

(1) $A:\ \nu\, a \cdot g^a\, \mu$ sendo $\mu = H(A\|B\|\pi)$

(2) $B:\ \nu\, b \cdot g^b \| H_1(\lambda)$ sendo $d = g^{a\,b}$ e $\lambda = A\|B\|g^a\,\mu\|g^b\|d\|\mu$

(3) $A:\ H_2(\lambda)$

A chave acordada é escolhida como $k = H_3(\lambda)$.

6.Criptografia Baseada na Identidade

Em 1984, Shamir[43] propôs o conceito de IBC ("Identity Based Cryptography") como um conjunto de técnicas assimétricas onde a identidade de um agente, descrita sob a forma de textos, endereços, mensagens, etc., pudesse ser usada como chave de cifras ou como meio de verificar uma assinatura.

Se compararmos esta abordagem com a "clássica" PKI vemos que, nesta última, cifrar mensagens ou verificar assinaturas é feito pelas chaves públicas. São estes items de informação que, perante os outros agentes, determinam a identidade do agente destinatário da mensagem cifrada ou autor de uma mensagem assinada.

Este tipo de identificação tem, porém, algumas desvantagens:

1. Nas técnicas PKI as chaves públicas são números gerados a partir de chaves privadas que, por seu turno, são valores escolhidos aleatoriamente. Por isso não têm qualquer significado fora do contexto onde são usadas. Por exemplo, vendo os bits de uma chave pública não se tem nenhuma indicação sobre quem é o legitimo titular dessa chave.
2. Exigem um item adicional (um certificado) para estabelecer a necessária relação entre chave e contexto de utilização. Esses certificados exigem, por seu turno, um enquadramento complexo para os gerar, distribuir e

[43] O "S" do RSA.

manter. Exige Autoridades de Certificação, exige um enquadramento jurídico para regular a sua actividade, exige agências reguladoras, etc.

A forma como as chaves privadas e públicas são geradas nas técnicas clássicas PKI faz com que a identificação contextual do agente (nome, endereços, fotografia, atributos de acesso, etc.) seja completamente independente da sua identificação criptográfica. A primeira é ditada pelo contexto informativo onde o agente interage; a segunda é ditada pela conveniência computacional da técnica criptográfica usada.

As técnicas IBC introduzem uma mudança radical: *a identificação contextual coincide com a identificação criptográfica*.

A identidade de um agente A, num contexto informativo, é essencialmente uma representação da legitimidade, perante terceiros, da titularidade de direitos. Por exemplo quando A assina digitalmente uma mensagem, a identidade descreve a autoria da mensagem; quando A se identifica perante uma qualquer instituição, a identidade descreve os direitos do agente na sua relação com essa instituição; quando uma mensagem destinada a A é cifrada, a identidade representa o direito de A de conhecer o seu conteúdo; etc.

Com esta visão da identidade, as consequências da abordagem IBC são diversas:

1. A identidade de um agente assume a forma de uma descrição textual de atributos[44] mas também dos direitos

[44] É costume classificar os atributos usados em identificação em três categorias: *atributos institucionais* (p.ex. nome, endereço, organização)

a que esta identidade está associada; por exemplo, pode conter um período de validade ou uma ACL ("access control list"). Uma identidade, usada num sistema de controlo de acessos, poderia ser

```
"Antonio Silva <asilva@qualquer.sitio> # 08/12/31 23:59 # universal: read, write"
```

que contém atributos institucionais, um limite temporal de validade e o tipo do acesso que permite.

2. A titularidade de uma identidade tem de ser estabelecida por uma gente de confiança (TA ou "trusted agent"). Essa titularidade, que inclui nomeadamente a capacidade para assinar as mensagens ou decifrar um criptograma, é verificada pelas técnicas criptográficas através do conhecimento que o agente titular tem de um determinado segredo; i.e uma *chave privada*.

   Desta forma, em qualquer esquema ou protocolo IBC, um passo essencial é o cálculo e distribuição, pelo TA, da chave privada associada à identidade do agente titular.

3. Adicionalmente o TA tem de fornecer ao contexto que usa a identidade de A (os agentes que verificam as assinaturas emitidas por A ou cifram as mensagens destinadas a A), garantias que foi realmente o TA o emissor da chave privada.

   Isso exige uma identificação publica do TA que, para tal, recorre a um par *chave pública+chave privada* no sentido clássico. Assim, a chave pública do TA vai ter de intervir em todos os esquemas públicos (verificação de assinaturas ou cifra de mensagens) onde intervenha a identidade.

que são determinados pelo contexto social onde o identificado está inserido, *atributos naturais* (p.ex. data e local de nascimento, identificação dos pais) que são determinados pelo seu nascimento e que correspondem ao acto de aquisição de personalidade jurídica, e *atributos físicos* tais como dados biométricos, fotografia, etc.

4. Ao contrário das técnicas clássicas, no IBC a identidade (chave pública) precede a chave privada. Por exemplo, uma mensagem pode ser cifrada sem que o destinatário disponha da capacidade para a decifrar; só quando tem necessidade de provar a titularidade da identidade é que o destinatário a apresenta perante o TA que, usando os mecanismos de validação que entender adequados, emite posteriormente a chave privada respectiva.

Dentro das técnicas IBC é conveniente identificar duas classes: a classe que agrupa as técnicas que têm como objectivo a autenticação (assinaturas, protocolos de identificação, acordos de chaves, etc.) e a classe das técnicas que estão orientadas à confidencialidade (cifras, principalmente).

As primeiras técnicas caem dentro do que se designa por **IBA ("Identity Based Autentication")**; as segundas caem dentro do que designa por **IBE ("Identity Based Encryption")**.

6.1 Técnicas IBC de Shamir

Como exemplos paradigmático dos princípios que acabámos de citar, vamos descrever duas técnicas (um esquema de asinaturas e um protocolo de acordo de chaves) que adaptam os princípios com que, em 1984, Shamir introduziu as técnicas IBC. Como seria de esperar do autor, estes protocolos são uma evolução do esquema de assinaturas RSA e usa os elementos comuns típicos de uma técnica RSA.

No esquema de assinaturas intervêm um "prover" $\mathcal{P}$ e um "verifier" $\mathcal{V}$. A identidade de $\mathcal{P}$ é representada pelo próprio símbolo.

Protocolo 27 : IBC de Shamir - Elementos comuns

Elementos comuns: É gerado um módulo RSA m e determinados
- um subgrupo primo $\Gamma \subseteq \mathbb{Z}_n^*$ de ordem r onde são feitas as multiplicações;
- duas funções de *hash*: $H : \Gamma \times \mathbb{B}^* \to \Gamma$ e *hash* $\mathsf{ID} : \mathbb{B}^* \to \Gamma \setminus 1$.

"Setup e Extract" T gera um par $\langle s, k \rangle$ de chaves RSA; torna m e k público. Distribui, por canal privado, a chave privada do "prover".

(1) $T : s \leftarrow \mathbb{Z}_\phi^* \quad ; \quad T : s \to k = s^{-1} \pmod{\phi}$
(2) $T : p \leftarrow \mathsf{ID}(\mathcal{P}) \quad ; \quad \lambda \leftarrow p^s$
(3) $\mathcal{P} : \lambda \leftarrow T.\lambda$

Protocolo 28 : Esquema de Assinaturas IBC de Shamir

Assinatura O "prover" constrói uma assinatura σ de um texto M

(1) $\mathcal{P}: \upsilon \leftarrow \mathbb{Z}_\phi^* \quad ; \quad \gamma \leftarrow \upsilon^k \quad ; \quad h \leftarrow H(\gamma, M)$
(2) $\mathcal{P}: \lambda \rightarrow \sigma = \gamma \,||\, \lambda\, \upsilon^h$ *assinatura*

Verificação o "verifier" $\mathcal{V}$ valida o triplo $\langle M, \sigma, \mathcal{P} \rangle$.

(1) $\mathcal{V}: \gamma, r \leftarrow \sigma \quad ; \quad h \leftarrow H(\gamma, M) \quad ; \quad p \leftarrow \mathsf{ID}(\mathcal{P})$
(2) $\mathcal{V}: r^k\, \gamma^{-h} \stackrel{?}{=} p$ *verificação*

JUSTIFICAÇÃO: Seja $p = \mathsf{ID}(\mathcal{P})$ e $h = H(\gamma, M)$. Como $\lambda = p^s$, pelo teorema RSA, tem-se $\lambda^k = p$; tem-se $r = \lambda\, \upsilon^h$, logo $r^k\, \gamma^{-h} = \lambda^k \cdot (\upsilon^k)^h\, \gamma^{-h} = p\, \gamma^h\, \gamma^{-h} = \mathsf{ID}(\mathcal{P})$.

Para o protocolo de acordo de chaves assume-se que existem dois agentes de identidades A e B que pretendem acordar num segredo comum usando canais abertos. O TA calcula as chaves privadas λ_A e λ_B associadas às identidades A e B e distribui-as aos agentes apropriados.

Cada instância do protocolo assenta ainda numa mensagem M pública que pode ser uma identidade de referência (por exemplo a identidade da técnica criptográfica usada, descrevendo textualmente a sua informação pública) ou uma mensagem específica sobre a qual o protocolo define um mecanismo de autenticação. No final cada agente não

só tem confiança no segredo acordado mas também tem uma prova de que o outro agente conhece a mensagem M (ou, pelo menos, o seu *hash*).

Protocolo 29 : Acordo de Chaves IBC de Shamir

Extact T calcula e distribui as chaves privadas de cada agente

$$\begin{array}{ll} (1) & T\colon p_A, p_B \leftarrow \mathsf{ID}(A)\,,\,\mathsf{ID}(B) \quad ; \quad \lambda_A, \lambda_B \leftarrow p_A^{-s}\,,\,p_B^{-s} \quad ; \quad M \rightarrow r = \mathsf{ID}(M) \\ (2) & A\colon \lambda_A \leftarrow T.\lambda_A \quad ; \quad B\colon \lambda_B \leftarrow T.\lambda_B \end{array}$$

Inicialização Cada agente gera um segredo e publicita a informação pública correspondente

$$\begin{array}{ll} (1) & A\colon a \leftarrow \mathbb{Z}_\phi^* \quad ; \quad B\colon b \leftarrow \mathbb{Z}_\phi^* \\ (2) & A\colon a, \lambda_A \rightarrow \sigma_A = \lambda_A\, r^a \quad ; \quad B\colon b, \lambda_B \rightarrow \sigma_B = \lambda_B\, r^b \end{array}$$

Geração do segredo:

$$(1) \quad A\colon \kappa_A \leftarrow \left(\sigma_B^k\, p_B\right)^a \quad ; \quad B\colon \kappa_B \leftarrow \left(\sigma_A^k\, p_A\right)^b$$

JUSTIFICAÇÃO: A definição das chaves privadas $lambda_A, \lambda_B$ e o teorema RSA asseguram que, em Γ,

$$\sigma_A^k\, p_A \;=\; \lambda_A^k\, r^{a\,k}\, p_A \;=\; p_A^{-k\,s}\, r^{a\,k}\, p_A \;=\; r^{a\,k}$$

e, de forma análoga, $\lambda_B^k\, p_B = r^{b\,k}$. Portanto,

$$\kappa_A \;=\; \left(\sigma_B^k\, p_B\right)^a \;=\; r^{a\,b\,k} \;=\; \left(\sigma_A^t\, p_A\right)^b \;=\; \kappa_B$$

Todos os protocolos de acordo de chaves requerem uma segunda fase em que ambos os agentes verificam que a chave acordada é a mesma. Para isso usam uma qualquer cifra simétrica para cifrar, com a chave que conhecem, informação que supostamente é comum.

Protocolo 30 : Acordo de Chaves de Shamir - Verificação

Elementos Comuns Um cifra simétrica $\langle E, D\rangle$

Informação pública $A\colon \kappa_A \to y_A = E(\kappa_A, \kappa_A) \quad ; \quad B\colon \kappa_B \to y_B = E(\kappa_B, \kappa_B)\,.$

Verificação $A\colon \kappa_A \;=?=\; D(\kappa_A, y_B) \quad ; \quad B\colon \kappa_B \;=?=\; D(\kappa_B, y_A)\,.$

6.2 Grupos "Gap" Diffie-Hellman (GDHG)

Seja $\langle \mathbb{G}, g, r \rangle$ um grupo Diffie-Hellman (abreviadamente **DHG** [45]); isto é um grupo cíclico onde o CDHP é considerado intratável.

Recordemos que, no CDHP é dado g^a e g^b e pretende-se calcular $g^{a\,b}$. Pode-se enfraquecer as exigências deste problema assumindo que também é dado um terceiro argumento g^c e pretende-se apenas decidir se c (que é desconhecido) coincide com $a\,b$ (que também é desconhecido).

Note-se que, devido às propriedades algébricas do grupo cíclico, verifica-se $c = a\,b$ se e só se $g^c = g^{a\,b}$. O problema em questão designa-se por "Decisão Diffie-Hellman" e pode, por isso, ser descrito como

Problema da Decisão Diffie-Hellman (DDHP)
Dados g^a, g^b *e* g^c *em* $\mathbb{G}$, *decidir se* $g^c = g^{a\,b}$.

Obviamente que DDHP é redutível a CDHP; de facto, se fornecermos g^a e g^b como *input* de um oráculo CDHP, ele force-nos $g^{a\,b}$ que pode ser comparado com g^c.

O inverso não é necessariamente verdade: existem grupos onde CDHP é comutacionalmente intratável e o DDHP não é intratável. De facto uma nova família de técnicas criptográficas surgiu nos últimos anos aproveitando este "gap" entre os dois problemas.

[45] Diffie-Hellman Group.

A definição do DDHP que apresentámos está demasiadamente informal: afinal o que significa "decidir"? Uma formalização probabilística do DDHP tem de tomar como referência uma determinada família de testes $\mathcal{T}$ e usar essa família para distinguir as linguagens que representam as duas situações que pretendemos diferenciar.

Concretamente, as linguagens em questão codificam triplos $\langle x, y, z\rangle \in G^3$ e definem-se do modo seguinte.

126 Definição
Dado um grupo DH $\langle \mathbb{G}, g, r\rangle$ *; então:*

- *A linguagem dos* **triplos Diffie-Hellman** *da órbita de* g *– representada por* $\mathsf{DHT}(g)$ *– é o conjunto de todas as bit-strings que codificam triplos* $\langle g^a, g^b, g^{a\,b}\rangle$ *quando* $a, b \in \mathbb{N}$ *percorrem todo o seu domínio.*
- *A linguagem dos* **pseudo triplos Diffie-Hellman** *da órbita de* g *– representada por* $\mathsf{PDHT}(g)$ *– é o conjunto de todas as bit-strings que codificam triplos* $\langle g^a, g^b, g^c\rangle$ *quando* $a, b, c \in \mathbb{N}$ *percorrem todo o seu domínio.*

A diferença entre estas duas linguagens está na terceira componente do triplo: em $\mathsf{DHT}(g)$ a componente $g^{a\,b}$ é determinada pelas duas primeiras componentes; na linguagem $\mathsf{PDHT}(g)$ a componente g^c é "livre". É óbvio que a segunda linguagem contém a primeira.

Agora pode-se dar uma forma mais precisa à definição do DDHP.

Problema da Decisão Diffie-Hellman Probabilística (DDHP)
Dados g *e um sistema de testes* $\mathcal{T}$*, encontrar um* $\mathcal{T}$*-diferenciador – ver definição* **??**, *pag.* **??** *– das linguagens* DHT(g) *e* PDHT(g).

Esta definição pode ser generalizada considerando, para além do grupo DH $\langle \mathbb{G}, g, r \rangle$, um outro grupo de torsão Γ, não necessariamente Diffie-Hellman.

127 Definição
Dado um grupo DH $\langle \mathbb{G}, g, r \rangle$ *e um grupo de torsão* Γ*; então:*

- *A linguagem dos* ***co-triplos Diffie-Hellman*** *da órbita de* g *e* Γ*– representada por* co-DHT(g, Γ) *– é o conjunto de todas as bit-strings que codificam triplos* $\langle g^a, \gamma, \gamma^a \rangle$ *quando* $a \in \mathbb{N}$ *e* $\gamma \in \Gamma$ *percorrem os respectivos domínios.*
- *A linguagem dos* ***pseudo co-triplos Diffie-Hellman*** *da órbita de* g *– representada por* co-PDHT(g, Γ) *– é o conjunto de todas as bit-strings que codificam triplos* $\langle g^a, \gamma, \gamma^c \rangle$ *quando* $a, c \in \mathbb{N}$ *e* $\gamma \in \Gamma$ *percorrem os respectivos domínios.*

Problema da Decisão Diffie-Hellman Probabilística Generalizado (co-DDHP)
Dados g, Γ *e um sistema de testes* $\mathcal{T}$*, encontrar um* $\mathcal{T}$*-diferenciador das linguagens* co-DHT(g, Γ) *e* co-PDHT(g, Γ).

É óbvio que esta segunda reformulação generaliza a primeira: basta considerar o caso em que Γ coincide com o grupo $\mathbb{G}$; neste caso, fazer γ percorrer Γ é equivalente a fazer b percorrer $\mathbb{N}$ e considerar $\gamma = g^b$.

□

Como já referimos os problemas da computação Diffie-Hellmam e da decisão Diffie-Hellman não são equivalentes. Em vários grupos DH (onde CDHP é intratável) existem algoritmos PPT para resolver o DDHP.

Em tais grupos existe uma "falha" (**gap**) entre os graus de complexidade computacional associados aos dois problemas; este "gap" pode ser explorado em técnicas criptográficas que baseiem o uso legítimo da técnica numa soluçaõ do DDHP e o uso ilegítimo (ataques) na solução do CDHP.

128 Definição
Um grupo DH $\langle \mathbb{G}, g, r \rangle$ *diz-se* ***gap-Diffie-Hellman*** *– abreviadamente* ***GDHG*** [46] *– para uma família de testes* $\mathcal{T}$ *quando DDHP tem solução para esse conjunto de testes.*

Sabendo que, neste grupos, CDHP não é computacionalmente viável mas DDHP é tratável, coloca-se a questão de como implementar esta decisão. Dois tipos de estratégia são conhecidos: as baseadas nos oráculos co-CDHP e as baseadas em emparelhamentos ("pairings").

[46] Gap Diffie-Hellman Group

co-CDHP Vamos assumir que se conhece um grupo cíclico Γ, com a mesma ordem r do que $\mathbb{G}$, para o qual co-CDHP é tratável. Formalmente assume-se

Existe um oráculo que, dados $\gamma \in \Gamma$ e $g^a \in G$ produz γ^a.

Decide-se se $\langle g^a, g^b, g^c \rangle$ é ou não um triplo DH em $\mathbb{G}$, da seguinte forma:

$\mathrm{DDHP}(g^a, g^b, g^c)$
(1) Gerar um $\gamma \in \Gamma$ aleatorio de ordem r,
(2) Usando o oráculo co-CDHP com γ e g^a, calcular γ^a,
(3) Usando o oráculo co-CDHP com γ^a e g^b, calcular $\gamma^{a\,b}$,
(4) Usando o oráculo co-CDHP com γ e g^c, calcular γ^c
(5) Aceitar $\langle g^a, g^b, g^c \rangle$ como triplo DH se e só se $\gamma^{a\,b} = \gamma^c$.

Algoritmo 1: Diferenciador DDHP derivado de um oráculo co-CDHP.

Note-se que, por γ ter a mesma ordem que o grupo, é um gerador de Γ e, portanto, verifica-se $\gamma^{a\,b} = \gamma^c$ sse $a\,b = c$ e, portanto, sse $g^{a\,b} = g^c$.

Emparelhamentos ("pairings") Vamos assumir que se conhece um grupo cíclico Γ, com a mesma ordem r de $\mathbb{G}$ e uma função $\mathbf{e}\colon G \times G \to \Gamma$ que satisfaz as seguintes propriedades:

(1) é *PT implementável*; i.e. existe um algoritmo PPT determinístico que implementa $\mathbf{e}$
(2) é *bilenear*; i.e., $\mathbf{e}(g^a, g^b) = \mathbf{e}(g, g)^{a\,b}$ para todos $a, b \in \mathbb{Z}_r$
(3) é *não-degenerado* no 1º argumento; i.e., $\mathbf{e}(g^a, g^c) = \mathbf{e}(g^b, g^c)$ sse $a = b$.

Uma tal função $\mathbf{e}$ designa-se por **emparelhamento**[47].

Dado um emparelhamento, decide-se se $\langle g^a, g^b, g^c\rangle$ é ou não um triplo DH em $\mathbb{G}$, da seguinte forma:

DDHP(g^a, g^b, g^c)
(1) Calcular $\mathbf{e}(g^a, g^b)$
(2) Calcular $\mathbf{e}(g^c, g)$
(3) Aceitar $\langle g^a, g^b, g^c\rangle$ como triplo DH se e só se os dois valores anteriores coincidirem.

Algoritmo 2: Diferenciador DDHP derivado de um emparelhamento.

[47] Uma definição mais geral de emparelhamento considera 3 grupos G, G', Γ e uma função $\mathbf{e}\colon G \times G' \to \Gamma$ que seja PT implementável, bilinear e não-degenerada no 1º argumento.

Note-se que, pela bilinearidade de $\mathbf{e}$, $\mathbf{e}(g^a, g^b) = \mathbf{e}(g, g)^{a\,b} = \mathbf{e}(g^{a\,b}, g)$. Então $\mathbf{e}(g^a, g^b) = \mathbf{e}(g^c, g)$ verifica-se sse $\mathbf{e}(g^{a\,b}, g) = \mathbf{e}(g^c, g)$ e, usando a não-degenerência do emparelhamento, essa igualdade verifica-se sse $a\,b = c$.

A técnica criptográfica mais simples, baseada em emparelhamentos, é um acordo tripartido de chaves: três agentes A, B e C usam um grupo cíclico e um emparelhamento para acordarem num segredo comum.

O protocolo que se segue é uma generalização óbvia do clássico acordo de chaves de Diffie-Hellman e, como este, está sujeito a ataques "homem-no-meio" se os agentes não estiverem autenticados mútuamente. Por isso é normalmente acompanhado com algum sistema de assinaturas que garanta essa autenticidade[48].

Os **elementos comuns** são um grupo DH $\langle \mathbb{G}, g, r \rangle$ e um emparelhamento $\mathbf{e}\colon G \times G \longrightarrow \Gamma$. Seja $\boldsymbol{\kappa}$ a função definida por $\boldsymbol{\kappa}(x, y, a) = \mathbf{e}(x, y)^a$.

[48] Por questões de eficiência implementacional, o esquema de assinaturas usa, geralmente, o mesmo grupo DH $\mathbb{G}$. Porém a versão que vamos aqui apresentar abstrai em relação à componente de autenticação mútua já que isso não é relevante para o uso dos emparelhamentos.

Protocolo 31 : Acordo Tripartido de Chaves (Joux)

Geração de chaves Cada uma dos agentes gera um segredo aleatório e publicita a chave pública correspondente.

(1) $A\colon a \leftarrow \mathbb{Z}_r^* \quad ; \quad B\colon b \leftarrow \mathbb{Z}_r^* \quad ; \quad C\colon c \leftarrow \mathbb{Z}_r^*$

(2) $A\colon a \rightarrow \sigma_A = g^a \quad ; \quad B\colon b \rightarrow \sigma_B = g^b \quad ; \quad C\colon c \rightarrow \sigma_C = g^c$

Geração do segredo comum Cada agente usa o emparelhamento e o seu segredo próprio

(1) $A\colon \kappa_A \leftarrow \boldsymbol{\kappa}(\sigma_B, \sigma_C, a) \quad ; \quad B\colon \kappa_B \leftarrow \boldsymbol{\kappa}(\sigma_A, \sigma_C, b) \quad ; \quad C\colon \kappa_C \leftarrow \boldsymbol{\kappa}(\sigma_B, \sigma_A, c)$

Justificação: Facilmente se verifica, usando a bilinearidade do emparelhamento, $\kappa_A = \kappa_B = \kappa_C = \mathbf{e}(g,g)^{a\,b\,c}$.

6.3 IBC em Grupos Gap Diffie-Hellman

A família de técnicas IBA caem dentro das técnicas implementáveis em grupos gap-DH e, de certa forma, o protocolo de Shamir é uma excepção. Normalmente as técnicas de IBA são definidas sobre grupos DH e, dentro destes, os grupos Gap-Diffie-Hellman têm uma importância particular.

Vamos assumir, no que se segue, que $\langle \mathbb{G}, g, r \rangle$ é um grupo gap-Diffie-Hellman equipado com um oráculo DDHP.

DDHP(g^a, g^b, g^c)
– *retorna o valor* 1 *se e só se os argumentos formam um triplo DH*

Não interessa a forma específica como é construído um tal oráculo: pode vir de um oráculo co-CDHP, de um emparelhamento ou de qualquer outra metodologia. Qualquer que seja o oráculo, convém atender ao resultado

129 Facto
Seja $q \neq 1$ *um elemento arbitrário de* G*;* $\langle g^a, q^b, q^c \rangle$ *é um triplo de Diffie-Hellman sse* $a\,b = c$.

Prova: Como g é gerador do grupo, existe sempre um $n \neq 0 \in \mathbb{Z}_r$ tal que $q = g^n$. Portanto o triplo $\langle g^a, q^b, q^c \rangle = \langle g^a, g^{n\,b}, q^{n\,c} \rangle$ é um DHT sse $n\,a\,b = n\,c$. Como $\mathbb{Z}_r$ não tem divisores de zero e $n \neq 0$, temos de concluir que $a\,b = c$.

Elementos Comuns

(1) Um grupo Gap-Diffie-Hellman $\langle \mathbb{G}, g, r \rangle$ munido de um oráculo DDHP
(2) Um gerador aleatório $\mathbb{Z}_r$ de elementos de $\mathbb{Z}_r$ uniformemente distribuídos
(3) Duas funções de *hash* $\mathrm{ID}: \mathbb{B}^* \rightarrow G \setminus 1$ e $H: G \times \mathbb{B}^* \rightarrow \mathbb{Z}_r$

Protocolo 32 : Assinatura IBA baseada em GDHG (Chon Cha & Hee Cheon)

Setup O TA gera o seu par de chaves: a privada s e a pública $\beta = g^s$; β é externamente autenticada.
(1) $T: s \leftarrow \mathbb{Z}_r \quad ; \quad s \rightarrow \beta = g^s$

Extract O TA calcula, e fornece ao "prover" $\mathcal{P}$, a chave privada λ correspondente à identidade $p = \mathrm{ID}(\mathcal{P})$. O "prover" $\mathcal{P}$ só aceita essa chave se decide, com sucesso, que $\langle \beta, p, \lambda \rangle$ é um DHT.
(1) $T: p \leftarrow \mathrm{ID}(\mathcal{P}) \quad ; \quad \lambda \leftarrow p^s$
(2) $\mathcal{P}: p \leftarrow \mathrm{ID}(\mathcal{P}) \quad ; \quad \mathrm{DDHP}(\beta \ , \ p \ , \ \lambda) \ =? \ 1$

Sign O "prover" $\mathcal{P}$ produz a assinatura σ a partir do texto M com a chave privada λ
(1) $\mathcal{P}: p \leftarrow \mathrm{ID}(\mathcal{P}) \quad ; \quad \upsilon, \gamma \leftarrow \mathbb{Z}_r \ , \ p^{\upsilon} \quad ; \quad h \leftarrow H(\gamma, M)$
(2) $\mathcal{P}: \lambda \rightarrow \sigma = \gamma \parallel \lambda^{\upsilon + h}$

Verify o "verifier" valida o triplo formado pelo texto M, a assinatura σ e a identidade $\mathcal{P}$
(1) $\mathcal{V}: p \leftarrow \mathrm{ID}(\mathcal{P}) \quad ; \quad \gamma, r \leftarrow \sigma \quad ; \quad h \leftarrow H(\gamma, M) \quad ; \quad t \leftarrow \gamma \cdot p^h$
(2) $\mathcal{V}: \mathrm{DDHP}(\beta \ , \ t \ , \ r) \ =? \ 1$

JUSTIFICAÇÃO: Seja $p = \mathrm{ID}(\mathcal{P})$. Para autenticação de λ atenda-se a que $\beta = g^s$ e $\lambda = p^s$; o *input* do oráculo DDHP é $\langle \beta, p, \lambda \rangle = \langle g^s, p, p^s \rangle$ que é um DHT (facto 129). Na verificação da assinatura, como $\gamma = p^v$, tem-se $t = p^{v+h}$; como $\lambda = p^s$, tem-se $r = \lambda^{v+h} = p^{s\,(v+h)}$. Os elementos fornecidos ao oráculo são $\langle g^s, p^{v+h}, p^{s\,(v+h)} \rangle$ e eles formam um DHT.

Este esquema tem associado um protocolo de identificação de conhecimento zero do qual deriva por uma transformação de Fiat-Shamir.

Protocolo 33 : Identificaçãp IBA baseada em GDHG

Setup & Extract Como no protocolo 32: o TA publica a sua chave pública β e computa e distribui a chave privada λ associada à identidade $\mathcal{P}$.

O "prover" $\mathcal{P}$ autentica previamente a sua chave privada resolvendo uma instância de DDHP.

Instância

(1) $\mathcal{P}\colon p \leftarrow \mathrm{ID}(\mathcal{P}) \;\; ; \;\; v \leftarrow \mathbb{Z}_r \;\; ; \;\; v \rightarrow \gamma = p^v$ *intensão*

(2) $\mathcal{V}\colon d \leftarrow \mathbb{Z}_r \;\; ; \;\; d \rightarrow d$ *desafio*

(3) $\mathcal{P}\colon \lambda \rightarrow r = \lambda^{v+d}$ *resposta*

(4) $\mathcal{V}\colon p \leftarrow \mathrm{ID}(\mathcal{P}) \;\; ; \;\; t \leftarrow \gamma \cdot p^d \;\; ; \;\; \mathrm{DDHP}(\beta\ ,\ t\ ,\ r) \stackrel{?}{=} 1$ *verificação*

JUSTIFICAÇÃO: A mesma que no protocolo 32 e derivada do facto que, no final do protocolo, se verifica $\beta = g^s$, $t = p^{v+d}$ e $r = p^{s\,(v+d)}$.

A segurança de qualquer um destes protocolos assenta, crucialmente, na crença de que o CDHP não pode ser resolvido neste grupo G. De facto um oráculo CDHP, ao receber os *inputs* $\langle \beta, p \rangle$ (sendo $p = \mathrm{ID}(\mathcal{P})$), calcula a chave privada $\lambda = p^s$.

Com o mesmo oráculo, após o passo (2) do protocolo de identificação, um intruso, mesmo sem conhecer λ, pode calcular a resposta correcta r. Dado que tem acesso ao β e consegue calcular o $t = \gamma \cdot p^d$, usa o oráculo CDHP sobre o par $\langle \beta, t \rangle$ e determina r; pode, deste modo, assumir a identidade do "prover" e substituí-lo no passo (3).

Um esquema de assinaturas bastante mais simples é designado por BLS e não está orientado à identidade. Este mecanismo BLS já foi usado, por nós, nos dois anteriores protocolos para autenticar a chave privada do "prover".

Protocolo 34 : BLS "short signatures" (Boneh et al.)

Geração de chaves O "prover" $\mathcal{P}$ gera uma chave privada s e a espectiva chave pública β que deve ser autenticada por um mecanismo externo (certificado).

(1) $\mathcal{P}$: $s \leftarrow \mathbb{Z}_r^*$; $s \rightarrow \beta = g^s$

Sign Assinatura σ da mensagem M

(1) $\mathcal{P}$: $h \leftarrow H(\beta, M)$; $s \rightarrow \sigma = h^s$

Verify verificar a validade da assinatura σ sobre o texto M com a chave pública β

(1) $\mathcal{V}$: $h \leftarrow H(\beta, M)$; $\mathrm{DDHP}(\beta\ ,\ h\ ,\ \sigma) =?= 1$

JUSTIFICAÇÃO: $\beta = g^s$, h e $\sigma = h^s$ formam um DHT.

Este esquema pode ser facilmente adaptado facilmente a assinaturas cegas.

Protocolo 35 : BLS em assinatura cega

Geração de chaves Como no protocolo 34

Sign O "verifier" cifra a mensagem previamente e recupera a assinatura original num 3º passo.

(1) $\mathcal{V}$: $\mu \leftarrow \mathbb{Z}_r \quad ; \quad h \leftarrow H(\beta, M) \quad ; \quad \mu \rightarrow y = g^{-\mu} \cdot h$ *– cifra*

(2) $\mathcal{P}$: $s \rightarrow x = y^s$ *– assinatura*

(3) $\mathcal{V}$: $\sigma \leftarrow x \cdot \beta^{\mu}$ *– recuperação*

Recuperação Como no protocolo 34

Justificação:

Temos $x = \left(g^{-\mu} \cdot h\right)^s = g^{-\mu\, s} \cdot h^s$. Uma vez que $\sigma = h^s$ e $\beta = g^s$, será $x = \beta^{-\mu} \cdot \sigma$. Desta forma $\beta^{\mu} \cdot x$ recupera a assinatura σ.

Adicionalmente $\mathcal{P}$ conhece $g^{-\mu} \cdot h$ e é capaz de calcular $\beta^{-\mu} \cdot \sigma$; nenhum destes termos lhe permite determinar μ, h ou σ.

6.4 IBE sobre Emparelhamentos

A definição de emparelhamento que se segue é uma pequena extensão da definição na págin 378

130 Definição

Seja $\langle \mathbb{G}, g, r \rangle$ *um grupo Diffie-Hellman e* G' *e* Γ *dois grupos cíclicos com a mesma ordem prima* r. *Seja* $\psi : \mathbb{G} \to G'$ *um isomorfismo de grupos com a propriedade* $\psi(g) \neq 1$.

Uma função $\mathbf{e} : G \times G_1 \to \Gamma$ *é um* **emparelhamento ("pairing")** *se for PPT implementável e*

- *não degenerada no* 1^o *argumento: isto é,* $\mathbf{e}(g^a, q) = \mathbf{e}(g^b, q)$ *implica* $a = b$, *e*
- *bilinear: isto é,* $\mathbf{e}(g^a, q^b) = \mathbf{e}(g, q)^{a\,b}$,

para todos $q \neq 1 \in G'$ *e* $a, b \in \mathbb{Z}_r$.

Note-se que num grupo cíclico de ordem prima r, qualquer elemento $p \neq 1$ é gerador do grupo[49]. Desta forma, para garantir que $\mathbf{e}(\cdot, \cdot)$ é não-degenerado basta garantir que $\gamma = \mathbf{e}(g, \psi(g)) \neq 1$ [50].

[49] Qualquer subgrupo de um grupo cíclico G tem uma ordem que divide a ordem do grupo. Se a ordem de G é um número primo, os únicos subgrupos de G são o próprio G e o subgrupo trivial $\{1\}$ formado só pelo elemento 1. A órbita de p só pode, portanto, ser esse subgrupo trivial (quando $p = 1$) ou, quando $p \neq 1$, ser todo o grupo G.

[50] Se $q \neq 1$ existe $n \neq 0$ tal que $q = \psi(g)^n$; então $\mathbf{e}(g^a, q) = \gamma^{a\,n}$ e $\mathbf{e}(g^b, q) = \gamma^{b\,n}$; sendo $\gamma^{a\,n} = \gamma^{b\,n}$, dado que $\gamma \neq 1$ é gerador do grupo Γ e $n \neq 0$ não é divisor de zero em $\mathbb{Z}_r$, será $a = b$.

A existência de um grupo DH onde está definido uma função emparelhamente permite estabelecer imediatamente um oráculo DDHP da forma que vimos anteriormente

– Dados $p, g^a, p^b, p^c \in \mathbb{G}$ *, se for* $p \neq 1$*, o oráculo decide se* $a\,b = c$ *.*

$\mathrm{DDHP}(g^a, p^b, p^c)$

decide $\mathbf{e}(g^a, \psi(p^b)) \stackrel{?}{=} \mathbf{e}(g, \psi(p^c))$

Algoritmo 3: Oráculo DDHP baseado em emparelhamentos.

Nota: $\mathbf{e}(g^a, \psi(p^b)) = \mathbf{e}(g^{a\,b}, \psi(p))$; também $\mathbf{e}(g, \psi(p^c)) = \mathbf{e}(g^c, \psi(p))$; como tem de ser $\psi(p) \neq 1$ e $\mathbf{e}$ é não-degenerado, terá de ser $a\,b = c$.

O poder dos emparelhamentos vai mais além desde que algumas condições adicionais de computabilidade e segurança sejam colocadas no grupo cíclico Γ e na função ψ. Note-se que a definição de um grupo DH exige que as operação de grupo seja PPT implementável. Isto faz com que a exponenciação seja PPT implementável mas não, obrigatoriamente, o logaritmo discreto.

Como não se impõe nos grupos auxiliares G' e Γ outras restrições para além de serem cíclicos de ordem r, a definição de emparelhamento não dá quaisquer garantias quanto à dificuldade do logaritmo discreto nestes grupos

como não existem garantias quanto à facilidade com que se implementam as operações de grupo e a exponenciação. Essas garantias têm de ser colocadas explicitamente em condições adicionais de computabilidade e segurança.

Primeiro as **condições de computabilidade**.

131 DEFINIÇÃO
Seja $\langle \mathbb{G}, g, r \rangle$ *um grupo DH onde está definido, como expresso na definição 130, um emparelhamento. Se for PPT implementável a função* $\boldsymbol{\kappa} \colon \mathbb{G} \times \mathbb{G} \times \mathbb{Z}_r \longrightarrow \Gamma$

$$\boldsymbol{\kappa} \ : \ \langle q, p, x \rangle \quad \mapsto \quad \mathbf{e}(q, \psi(p))^x \qquad com \quad q, p \in \mathbb{G} \ , \ x \in \mathbb{Z}_r \tag{118}$$

o grupo diz-se ***bilinear DH***.

O problema que está na base das várias aplicações criptográficas dos emparelhamentos é o já referido protocolo de Joux para o acordo tripartido de chaves. Recordemos, que o protocolo baseava-se em três agentes A, B e C, que são titulares de chaves privadas a, b e c, e que procuram acordar numa chave comum κ

A chave acordada, dita P3K ("pairings triple key"), é determinado por um $p \neq 1 \in \mathbb{G}$ e pelos três segredos $a, b, c \in \mathbb{Z}_r$. É um item da forma $\kappa = \gamma^{a\,b\,c}$, em que $\gamma = \mathbf{e}(g, \psi(p)) = \boldsymbol{\kappa}(g, p, 1)$, que é calculada de forma diferente consoante o conhecimento de cada agente.

- O agente C conhece c e conhece também $\langle g^a, p^b \rangle$ ou $\langle g^b, p^a \rangle$

$C: \kappa \leftarrow \boldsymbol{\kappa}(g^a, p^b, c)$ ou $C: \kappa \leftarrow \boldsymbol{\kappa}(g^b, p^a, c)$

- O agente B conhece b e também $\langle g^a, p^c \rangle$ ou $\langle g^c, p^a \rangle$

$B: \kappa \leftarrow \boldsymbol{\kappa}(g^a, p^c, b)$ ou $B: \kappa \leftarrow \boldsymbol{\kappa}(g^c, p^a, b)$

- O agente A conhece a e conhece $\langle g^b, p^c \rangle$ ou $\langle g^c, p^b \rangle$

$A: \kappa \leftarrow \boldsymbol{\kappa}(g^b, p^c, a)$ ou $A: \kappa \leftarrow \boldsymbol{\kappa}(g^c, p^b, a)$

As diferentes técnicas criptográficas assentes no cálculo de chaves P3K baseiam a sua correcção na capacidade de calcular a mesma chave de forma diferente por vários agentes consoante os segredos que possuem. Por exemplo numa cifra, a chave será calculada na cifragem com a identidade do destinatário e com um segredo de sessão aleatório e, na de-cifragem, com a chave privada e com a exponenciação do segredo de sessão.

A propriedade algébrica que permite assegurar essa correcção, e que deriva imediatamente da definição (118), é

$$\boldsymbol{\kappa}(g^a, p^b, c) \;=\; \boldsymbol{\kappa}(g^x, p^y, z) \qquad \text{sse} \qquad a\,b\,c \;=\; x\,y\,z \tag{119}$$

Uma consequência desta propriedade, que por si só, é suficiente para assegurar a correcção é

132 Facto

O valor de $\boldsymbol{\kappa}(g^a, p^b, c)$ *é invariante em relação a qualquer permutação que for feita no tuplo* $\langle a, b, c\rangle$.

□

O cálculo da chave P3K é possível quando o agente que a calcula conhece um dos segredos (a, b ou c). Note-se que a função $\boldsymbol{\kappa}$ toma dois argumentos no grupo $\mathbb{G}$ mas um terceiro argumento no "domínio dos segredos" $\mathbb{Z}_r$. As **condições de segurança** que iremos estabelecer impõe que esta condição, para além de suficiente, tem de ser necessária; isto é, não deve ser possível calcular a chave κ sem conhecer um dos segredos.

Assim, seja $\langle \mathbb{G}, g, r\rangle$ um grupo bilinear DH (segundo a definição 130 e a definição 131); neste contexto existem três problemas que permitem exprimir a segurança.

Problema da Computação DH Bilinear (BCDHP)
Dados $p \neq 1$ *e pares* $\langle g^a, p^a\rangle, \langle g^b, p^b\rangle, \langle g^c, p^c\rangle$, *determinar* $\boldsymbol{\kappa}(g^a, p^b, c)$.

Este problema tem uma versão sob forma de decisão

Problema da Decisão DH Bilinear (BDDHP)
Dados $p \neq 1$ *e pares* $\langle g^a, p^a\rangle, \langle g^b, p^b\rangle, \langle g^c, p^c\rangle$, *distinguir* $\boldsymbol{\kappa}(g^a, p^b, c)$ *de* $\boldsymbol{\kappa}(g^a, p^b, x)$, *com* $x \in \mathbb{Z}_r$ *aleatório.*

Uma variante de BCDHP considera um número finito de elementos de $\mathbb{G}$ gerados pelo expoente desconhecido a a partir de bases g e p. Para cada função $f : \mathbb{Z}_r \longrightarrow \mathbb{Z}_r$ e limite n define-se

Problema da Computação DH Bilinear f, n (f, n-BCDHP)

Dados n *pares* $\langle g^{a^i}, p^{a^i} \rangle \in \mathbb{G}^2$, *com* $i = 0 \ldots n-1$, *determinar* $\boldsymbol{\kappa}(g, p, f(a))$.

□

Como exemplo de aplicação imediata destas noções vamos considerar o esquema de cifra de Boneh & Franklin (2001) que veio reactivar o moderno interesse em IBC.

Os seguintes elementos ocorrem em todos os protocolos derivados do protocolo básico de Boneh& Franklin.

Protocolo 36 : Elementos Comuns em IBC do tipo Boneh & Franklin

(1) O espaço dos textos $\mathbb{B}^t$.
(2) Um grupo de DH bilinear $\langle \mathbb{G}, g, r \rangle$ onde é BCDHP é intratável.
(3) Uma funçao de *hash* para representação de identidade: $\mathrm{ID} : \mathbb{B}^* \longrightarrow \mathbb{G} \setminus 1$.
(4) Funções de *hash*: $H : \mathbb{B}^* \longrightarrow \mathbb{B}^t$ e $H_r : \mathbb{B}^* \longrightarrow \mathbb{Z}_r$.
(5) Função *hash* de conversão $f : \Gamma \longrightarrow \mathbb{B}^t$.

Os **agentes** em causa são o "agente de confiança" T, o "encryptor" $\mathcal{E}$ e "decryptor" $\mathcal{D}$. O **objectivo** de $\mathcal{E}$ é cifrar um texto m (limitado a t bits) usando a identidade $\mathcal{D}$ como chave pública.

Protocolo 37 : Esquema IBE de Boneh & Franklin

"Setup" TA gera a sua chave privada s e respectiva chave pública $\beta = g^s$ que é certificada externamente

(1) $T\colon s \leftarrow \mathbb{Z}_r^* \quad ; \quad s \rightarrow \beta = g^s$

"Extract" TA determina a representação da identidade do "decryptor" e fornece-lhe a respectiva chave privada por canal fechado; $\mathcal{D}$ autentica a chave que recebe usando o oráculo DDHP gerado pelo emparelhamento.

(1) $T\colon d \leftarrow \mathsf{ID}(\mathcal{D}) \quad ; \quad \lambda \leftarrow d^s$

(2) $\mathcal{D}\colon \lambda \leftarrow T.\lambda \quad ; \quad d \leftarrow \mathsf{ID}(\mathcal{D}) \quad ; \quad \mathrm{DDHP}(\beta, d, \lambda) \stackrel{?}{=} 1$

Cifra/"Encryption" O texto $m \in \mathbb{B}^t$ é cifrado com a identidade $\mathcal{D}$ e produz o criptograma σ

(1) $\mathcal{E}\colon \upsilon \leftarrow \mathbb{B}^t \quad ; \quad a \leftarrow H_r(\upsilon \| m)$

(2) $\mathcal{E}\colon d \leftarrow \mathsf{ID}(\mathcal{D}) \quad ; \quad \mu \leftarrow \boldsymbol{\kappa}(\beta, d, a)$

(3) $\mathcal{E}\colon a, \upsilon, \mu, m \rightarrow \sigma = g^a \,\|\, \upsilon \oplus f(\mu) \,\|\, m \oplus H(\upsilon)$

Decifra/"Decryption" Com a chave privada λ e o criptograma $\sigma = \gamma \| \upsilon' \| m'$ recupera-se o texto m

(1) $\mathcal{D}\colon \gamma, \upsilon', m' \leftarrow \sigma \quad ; \quad \mu \leftarrow \boldsymbol{\kappa}(\gamma, \lambda, 1)$

(2) $\mathcal{D}\colon \upsilon \leftarrow \upsilon' \oplus f(\mu) \quad ; \quad m \leftarrow m' \oplus H(\upsilon)$

(3) $\mathcal{D}\colon a \leftarrow H_r(\upsilon \| m) \quad ; \quad \gamma \stackrel{?}{=} g^a$

A correcção deste protocolo assenta na constação de que o valor μ usado no esquema de cifra coincide com o valor

μ usado no esquema de decifra. Na cifra usa-se o triplo de valores $\langle g^s, d^1, a\rangle$; no esquema de decifra usa-se os valores $\langle g^a, d^s, 1\rangle$. Ambos derivam de permutações do mesmo triplo de "segredos" $\langle s, a, 1\rangle$; portanto o valor de μ é o mesmo em ambos os casos.

Note-se que o último passo no esquema de decifra existe apenas como confirmação de que o protocolo foi correctamente executado. O esquema de decifra permite recuperar todos os segredos (excepto a chave privada) gerados pelo esquema de cifra: o valor aleatório υ e a sua projecção a no domínio $\mathbb{Z}_r$ e o texto m; o teste $\gamma =? g^a$ permite verificar se o valor recuperado a coincide com o valor que foi usado para gerar γ no criptograma.

□

A segurança deste protocolo assenta na intratabilidade do BCDHP (na incapacidade de recuperar μ do conhecimento de $\beta = g^s$, $\gamma = g^a$ e d^1 apenas) mas, crucialmente, da forma como as quatro funções de *hash* usadas (ID, H, H_r e f) preservam, no *output*, a aleatoriedade do *input*.

Um modelo de segurança, onde se assume que todas as funções de *hash*, quando sujeitas a *inputs* aleatórios, se comportam como geradores aleatórios, designa-se por **modelo de oráculo aleatório ("random oracle model" ou ROM)**.

No ROM é relativamente fácil demonstrar a segurança do protocolo acima. No entanto a adopção de um tal modelo é sempre questionável; a alternativa seria concretizar as funções de *hash* em esquemas computacionais específicos e estudar o comportamento do esquema IBC com essas concretizações.

Um modelo de segurança que assume apenas a intratabilidade computacional dos diferentes problemas (CDHP, BCDHP, DLP, etc.) e não impõe quaisquer assumpções adicionais, designa-se por **modelo standard**.

A presença de muitas funções de *hash* abstractas levanta a impossibilidade de aplicar o modelo standard nas provas de segurança deste protocolo. Idealmente um esquema deveria reduzir ao mínimo o número dessas funções para tentar aproximar, tanto quanto possível, o seu modelo de segurança do modelo standard.

Por este motivo o protocolo 37 não é facilmente analisável num modelo standard de segurança.

□

Um outro aspecto a ter em conta, na avaliação de esquemas e protocolos em IBC, é a forma como as chaves privadas são geradas e distribuídas.

Em todos os protocolos IBC vistos até ao momento, as chaves privadas são gerados por um "trusted agent" que, posteriormente, as distribui aos seus titulares usando canais privados. Isto significa que, numa comunidade de agentes dependente de um único TA, existe um agente privilegiado que detém o conhecimento de todas as chaves privadas: ele possui um **depósito em confiança** (**"escrow"**) das chaves privadas de toda a comunidade.

Com esse depósito, o TA pode definir as políticas que entender para a distribuição das chaves; nomeadamente ele pode (por força ou por ser vítima de ataque) revelar uma chave privada a alguém que não é titular da mesma. Esta

situação é, obviamente, indesejável; a comunidade de agentes pode achar que um tal poder não pode ser depositado num único agente.

Por isso as técnicas IBC são também avaliadas pelo grau de **"escrow-freeness"** que apresentam. Por exemplo, para evitar a dependência em relação ao depósito de confiança, um protocolo pode distribuir a geração de chaves por várias fontes de confiança de forma a que nenhuma delas, individualmente, seja capaz de reconstruir qualquer chave.

Claramente o protocolo 37 avalia mal, não só em termos de modelo standard de segurança, mas também em "escrow-freeness". Veremos em seguida algumas alternativas.

□

Um terceiro aspecto diz respeito à forma como a identidade de um agente, expressa genericamente numa string de bits, é convertida em um objecto de um dos domínios que fazem parte dos elementos comuns da técnica criptográfica. Esse objecto designamos por **representante** da identidade.

Esta questão está associada, normalmente, à forma como a chave privada de um agente é gerada a partir da informação privada das fontes de confiança e do representante da identidade.

No contexto de técnicas baseadas em grupos DH, todos os protocolos apresentados até ao momento (nomeadamente o IBE de Boneh & Franklin) encaram estes dois aspectos sempre da mesma forma:

- Cada identidade $I \in \mathbb{B}^*$ é representada por um elemento p do grupo cíclico $\mathbb{G} \setminus 1$. Esse representante constrói-se como $p = \mathsf{ID}(I)$ sendo $\mathsf{ID}: \mathbb{B}^* \rightarrow \mathbb{G} \setminus 1$ uma função de *hash*.
- Existe uma única fonte de confiança TA que tem uma chave privada no "domínio dos segredos" $s \in \mathbb{Z}_r$. A chave pública correspondente é da forma $\beta = g^s$.
- A chave privada λ_I do agente titular da idendidade I é dada por $\lambda_I = p^s$.

Este mecanismo concretiza-se nas fases "setup" e "extract" desta família de técnicas

Protocolo 38 : Geração de Chaves do Tipo I (Boneh & Franklin)

"Setup" O TA gera a sua chave privada s e publicita a respectiva chave pública $\beta = g^s$

(1) $T: s \leftarrow \mathbb{Z}_r \quad ; \quad s \rightarrow \beta = g^s$

"Extract" O TA calcula a chave privada de I e envia-a por canal privado para o seu titular.

(1) $T: p \leftarrow \mathsf{ID}(I) \quad ; \quad \lambda \leftarrow p^s$

(2) $I: \lambda \leftarrow T.\lambda$ *– o agente I recolhe o conhecimento λ de T por canal privado*

Esta construção permite uma versão "escrow-free". Para isso considera-se que o TA está distribuído por várias fontes de confiança $T_1, T_2, \ldots, T_d$ cada uma com a sua chave privada $s_1, s_2, \ldots, s_n$.

Cada T_k vai proceder exactamente como se fosse um único TA no esquema anterior. No entanto, individualmente, cada um desses agentes produz apenas parte da chave pública e parte da chave privada. Um utilizador (da chave

pública ou da chave privada) tem de as reconstruir a chave que necessita a partir das várias componentes fornecidas pelas fontes.

Protocolo 39 : Geração de Chaves do Tipo I (versão "escrow-free")

"Setup" Cada T_k gera a sua chave privada s_k e publicita a respectiva componente da chave pública $\beta_k = g^{s_k}$

(1) T_k: $s_k \leftarrow \mathbb{Z}_r$; $s_k \rightarrow \beta_k = g^{s_k}$ – *para todo* $k = 1 \ldots d$

"Extract" Cada T_k calcula um factor λ_k da chave privada λ

(1) T_k: $p \leftarrow \mathsf{ID}(I)$; $\lambda_k \leftarrow p^{s_k}$ – *para todo* $k = 1 \ldots d$

(2) I: $\lambda_k \leftarrow T_k.\lambda_k$ – *para todo* $k = 1 \ldots d$

Construção de chaves Usando uma "máscara" $\overline{b} = b_1 \| b_2 \| \ldots \| b_d \in \mathbb{Z}_r^d$, o público $*$ reconstrói a chave pública β e I reconstrói a chave privada λ.

(1) $*$: $\beta \leftarrow (\beta_1)^{b_1} \ (\beta_2)^{b_2} \ \ldots \ (\beta_d)^{b_d}$

(2) I: $\lambda \leftarrow (\lambda_1)^{b_1} \ (\lambda_2)^{b_2} \ \ldots \ (\lambda_d)^{b_d}$

Se definirmos $s = \sum_{k=1}^{d} b_k \, s_k$, vê-se facilmente que $\beta = g^s$ e que $\lambda = p^s$. Portanto este esquema comporta-se exactamente como o esquema original de Boneh & Franklin com a diferença de que a "chave privada" s nunca chega a existir; é, de certa forma, "virtual".

Cada uma das fontes de confiança T_k conhece uma parte de s mas não conhece a chave toda. Desta forma, a

menos que exista um conluio completo entre as várias fontes de confiança, nenhum dos T_k individualmente conhece a chave privada λ e, por isso, não a pode divulgar (mesmo que queira).

Adicionalmente, se existir evidencia que T_k foi atacado ou divulgou informação privilegiada, as respectivas componentes β_k e λ_k podem ser retiradas da construção das chaves β e λ preservando a relação entre essas chaves. Esta é a função da "máscara" $b_1, \ldots, b_d$: escolher as fontes de confiança que fornecem a informação usada na reconstrução de β e λ e definir um "peso" para cada uma dessas componentes.

Um protocolo que esconda a máscara $\overline{b}$ das fontes de confiança, é "escrow-free" e "anti-conluio"; isto porque, mesmo que as fontes estejam em conluio completo, não podem prever a máscara e, assim, não podem reconstruir λ. Obviamente que um tal protocolo exige que a máscara $\overline{b}$ seja um segredo comum apenas dos intervenientes do protocolo e tal implica, normalmente, o uso prévio de um protocolo de acordo de chaves.

□

Uma abordagem alternativa mapeia a identidade I, não no grupo $\mathbb{G}$, mas no mesmo domínio $\mathbb{Z}_r$ que contém as chaves privadas das eventuais fontes de confiança. A função de *hash* usada tem agora a forma $\mathsf{ID}: \mathbb{B}^* \longrightarrow \mathbb{Z}_r$ e o representante é também construído como $p = \mathsf{ID}(I)$.

Este solução presta-se à construção de funções polinomiais $q(p, s)$ que "misturam" a identidade p com a chave privada s de um TA. Com tais funções, a exponenciais $g^{q(p,s)}$ podem ser reconstruídas a partir do conhecimento de p e das potências $\beta_0 = g, \beta_1 = g^s, \beta_2 = g^{s^2}$, etc.

Por exemplo, um polinómio

$$q(p,s) \;=\; p^2 + s\,p + s^2$$

permite construir um valor $\mu = g^{f(p,s)}$ como $\mu \;=\; g^{p^2}\;(g^s)^p\;g^{s^2}$. Se os valores $\beta_0 = g, \beta_1 = g^s$ e $\beta_2 = g^{s^2}$ constituírem a chave pública, o conhecimento de p permite o cálculo público de

$$\mu \;=\; g^{q(p,s)} \;=\; \beta_0^{p^2} \cdot \beta_1^p \cdot \beta_2$$

Esta abordagem está inerente ao segundo tipo de IBC baseado em grupos Diffie-Helmman bilineares. Os elementos comuns são, essencialmente, os mesmos que estão descritos no protocolo 36. A única diferença é a função de *hash* ID que, aqui, tem $\mathbb{Z}_r$ como contradomínio.

Protocolo 40 : Elementos Comuns em IBC do tipo Sakai & Kasahara

(1) O espaço dos textos $\mathbb{B}^t$.
(2) Um grupo de DH bilinear $\langle \mathbb{G}, g, r \rangle$ onde é BCDHP é intratável.
(3) Uma funçao de *hash* para representação de identidade: ID : $\mathbb{B}^* \longrightarrow \mathbb{Z}_r$
(4) Funções de *hash*: $H : \mathbb{B}^* \longrightarrow \mathbb{B}^t$ e $H_r : \mathbb{B}^* \longrightarrow \mathbb{Z}_r$.
(5) Função *hash* de conversão $f : \Gamma \longrightarrow \mathbb{B}^t$.

A geração de chaves que vamos apresentar mistura o segredo s e a identidade p num monómio $(p + s)$.

Protocolo 41 : Geração de Chaves do Tipo II (Sakai & Kasahara)

"Setup" Como no tipo I

(1) $T\colon s \leftarrow \mathbb{Z}_r^* \quad ; \quad s \rightarrow \beta = g^s$

"Extract" A chave privada λ é gerada e comunicada por canal privado.

(1) $T\colon p \leftarrow \mathsf{ID}(I) \quad ; \quad z \leftarrow (p+s)^{-1} \quad ; \quad \lambda \leftarrow g^z$

(2) $I\colon \lambda \leftarrow T.\lambda$

Protocolo 42 : Esquema IBE (Sakai & Kasahara)

"Setup&Extract" Geração de chaves do tipo II

Cifra/ "Encryption" O texto $m \in \mathbb{B}^t$ é cifrado com a identidade $\mathcal{D}$ e produz o criptograma σ

(1) $\mathcal{E}\colon \upsilon \leftarrow \mathbb{B}^t \quad ; \quad a \leftarrow H_r(\upsilon \| m) \quad ; \quad \mu \leftarrow \boldsymbol{\kappa}(g,g,a)$

(2) $\mathcal{E}\colon d \leftarrow \mathsf{ID}(\mathcal{D}) \quad ; \quad \gamma \leftarrow (\beta\, g^d)^a$

(3) $\mathcal{E}\colon \upsilon, \mu, m \rightarrow \sigma = \gamma \,\|\, \upsilon \oplus f(\mu) \,\|\, m \oplus H(\upsilon)$

Decifra/ "Decryption" Com a chave privada λ e o criptograma $\sigma = \gamma \| \upsilon' \| m'$ recupera-se o texto m

(1) $\mathcal{D}\colon \gamma, \upsilon', m' \leftarrow \sigma \quad ; \quad \mu \leftarrow \boldsymbol{\kappa}(\gamma, \lambda, 1)$

(2) $\mathcal{D}\colon \upsilon \leftarrow \upsilon' \oplus f(\mu) \quad ; \quad m \leftarrow m' \oplus H(\upsilon)$ – *recuperação*

(3) $\mathcal{D}\colon a \leftarrow H_r(\upsilon \| m) \quad ; \quad d \leftarrow \mathsf{ID}(\mathcal{D}) \quad ; \quad \gamma \stackrel{?}{=} (\beta\, g^d)^a$ – *verificação*

O protocolo de IBE originário de Sakai & Kasahara (2003) que é essencialmente o protocolo de Boneh & Franklin adaptado a esta configuração de chaves. Os agentes e o objectivo são os mesmos do protocolo 37 de Boneh & Franklin.

A correcção deriva do facto de ser $\gamma = g^{(s+d)\,a}$, de ser $\lambda = g^{(s+d)^{-1}}$ e ser $\mu = \boldsymbol{\kappa}(g, g, a) = \boldsymbol{\kappa}(g^{(s+d)\,a}, g^{(s+d)^{-1}}, 1)$. A segurança deriva do facto de ser intratável determinar $g^{(s+d)^{-1}}$ mesmo que se conheça g^s e g^d.

Este esquema pode ser generalizado, substituindo o monómio $(s+d)$ por outro qualquer polinómio $q(s,d)$. Para isso a chave pública é agora formada pelos vários $\beta_i = g^{s^i}$, com i desde 0 até ao grau do polinómio. A chave privada seria $\lambda = g^{q(s,d)^{-1}}$ e o cálculo de γ reconstrói $g^{a\,p(s,d)}$ a partir dos vários β_i e das potências g^{d^i}.

□

Outra generalização substitui $\boldsymbol{\kappa}$, apresentada em (119), por uma construção algébrica chamada **co-emparelhamento**.

133 DEFINIÇÃO
Sejam $\langle \mathbb{G}, g, r\rangle$ *e* $\langle \Gamma, h, r\rangle$ *dois grupos DH com a mesma ordem prima* r. *Um* **co-emparelhamento** *é uma função PPT implementável* $\mathbf{c}\colon \mathbb{G} \times \Gamma \to \Gamma$ *que é não-degenerada (* $\mathbf{c}(g,h) \neq 1$ *) e bilinear (* $\mathbf{c}(g^a, \gamma) = \gamma^a$, *para todo* $a \in \mathbb{Z}_r$ *e* $\gamma \in \Gamma$ *).*

Um co-emparelhamento produz, imediatamente, um oráculo DDHP.

– Dados $g^a, p^b, p^c \in \mathbb{G}$, se for $p \neq 1$, o oráculo decide se $a\,b = c$.

$\mathrm{DDHP}(g^a, p^b, p^c)$

$\gamma \leftarrow \mathbf{c}(g^a, h)$
decide $\mathbf{c}(p^c, h) \;=?=\; \mathbf{c}(p^b, \gamma)$

Algoritmo 4: Oráculo DDHP baseado em co-emparelhamentos.

No esquema de geração de chaves tipo II (protocolo 41), λ passa a ser um elemento do grupo Γ.

Protocolo 43 : Geração de Chaves tipo II - variante co-emparelhamento

"Setup" Como no tipo II, protocolo 41

"Extract" A chave privada λ é gerada e comunicada por canal privado.

(1) $T: d \leftarrow \mathrm{ID}(I) \quad ; \quad z \leftarrow (s+d)^{-1} \quad ; \quad \lambda \leftarrow h^z$

(2) $I: \lambda \leftarrow T.\lambda$

No esquema IBE de decifra, μ passa a ser gerado pela função co-emparelhamento $\mathbf{c}$.

Protocolo 44 : Esquema IBE (Sakai & Kasahara) - variante co-emparelhamento

"Setup&Extract" protocolo 43

Cifra/ "Encryption" O texto $m \in \mathbb{B}^t$ é cifrado com a identidade $\mathcal{D}$ e produz o criptograma σ

(1) $\mathcal{E}: \upsilon \leftarrow \mathbb{B}^t \;\; ; \;\; a \leftarrow H_r(\upsilon \| m) \;\; ; \;\; \mu \leftarrow h^a$

(2) $\mathcal{E}: d \leftarrow \mathsf{ID}(\mathcal{D}) \;\; ; \;\; \gamma \leftarrow (\beta\, g^d)^a$

(3) $\mathcal{E}: \upsilon, \mu \;\rightarrow\; \sigma = \gamma \,\|\, \upsilon \oplus f(\mu) \,\|\, m \oplus H(\upsilon)$

Decifra/"Decryption" Com a chave privada λ e o criptograma $\sigma = \gamma \| \upsilon' \| m'$ recupera-se o texto m

(1) $\mathcal{D}: \gamma, \upsilon', m' \leftarrow \sigma \;\; ; \;\; \mu \leftarrow \mathbf{c}(\gamma, \lambda)$

(2) $\mathcal{D}: \upsilon \leftarrow \upsilon' \oplus f(\mu) \;\; ; \;\; m \leftarrow m' \oplus H(\upsilon)$ – *recuperação*

(3) $\mathcal{D}: a \leftarrow H_r(\upsilon \| m) \;\; ; \;\; \gamma \;=?=\; (\beta\, g^d)^a$ – *verificação*

Justificação A correcção desta versão do esquema deriva da bilinearidade da função de co-emparelhamento:

$$\mu \;=\; \mathbf{c}(\gamma, \lambda) \;=\; \mathbf{c}(g^{a\,(s+d)}, h^{(s+d)^{-1}}) \;=\; h^a$$

□

As cifras assimétricas têm uma complexidade computacional que torna inviável usá-las com mensagens grandes. Por

isso são usadas quase só em mensagens muito curtas; tipicamente, em chaves.

Essas chaves, assim cifradas, são enviadas a um destinatário e constituiem, desta forma, segredos que o gerador e o destinatário partilham. Finalmente as chaves partilhadas podem ser usadas numa cifra simétrica que, por ser computacionalmente muito eficiente, tem capacidade para processar mensagens de tamanho arbitrário.

Este mecanismo é designado por **cifragem híbrida**. Tradicionalmente o uso de uma cifra híbrida por um agente E, que quer enviar a um outro agente D uma mensagem m cifrada, assume a existência de:

- Uma cifra assimétrica (**Cifra** e **Decifra**) e uma cifra simétrica (**SimCifra** e **SimDecifra**).
- Uma par de chaves (ek, dk), pública e privada, para a cifra assimétrica.

Protocolo 45 : Cifra Híbrida

Cifra Cifrar um texto m de tamanho arbitrário

(1) E gera uma chave aleatória k, determina $\sigma = \textbf{Cifra}(k, \text{ek})$ e torna público este valor

(2) E usa k para cifrar o texto m com a cifra simétrica: faz $y = \textbf{SimCifra}(m.k)$ e torna-o público.

Decifra D recupera a chave k e o texto m

(1) D recupera k com a chave privada: $k \leftarrow \textbf{Decifra}(\sigma, \text{dk})$.

(2) D recupera, com k, o texto m: $m \leftarrow \textbf{SimDecifra}(y, k)$.

A análise de segurança deste protocolo básico mostra-se mais simples que separar-mos as primitivas (1) da cifragem e decifragem num único esquema e juntarmos as primitivas (2) destes mesmo passos num segundo o esquema.

Esta abordagem designa-se por **KEM-DEM**.

A primeira componente, o **Key Encapsulation Mechanism (KEM)**, corresponde *grosso modo*, ao passo de geração do par de chaves $(\mathsf{ek}, \mathsf{dk})$, ao passo de geração do segredo k e a sua cifra ("encapsulamento") com a chave pública ek, e ao passo de extracção ("desencapsulamento") de k usando a chave privada dk.

Na segunda componente intervém o texto m (os "dados"); designa-se por **Data Encapsulation Mechanism (DEM)** e é descrito pela cifragem do texto m (e sua recuperação) usando, num esquema de cifra/decifra simétrica, a chave k gerada e recuperada pelo KEM.

Uma característica importante dos esquemas **KEM** actuais é que a **geração** da chave k e o seu **encapsulamento**, é feita globalmente num único passo lógico. Isto contrasta com o esquema clássico das cifras híbridas onde a geração de k e a sua cifragem são algoritmos separados.

Outra característica importante reside no facto de a componente **DEM** poder ser completamente independente da componente KEM. Nomeadamente a segurança da DEM é completamente independente da segurança da KEM; o único ponto que têm em comum é a chave k cujo tamanho força alguma dependência entre as duas componentes. Por isso, e dado que a segurança do DEM é essencialmente a de uma cifra simétrica, pode-se abstrair o estudo desta técnica só ao estudo do KEM.

Protocolo 46 : Modelo KEM-DEM

- O **Key Encapsulation Mechanism (KEM)** é formado por 3 algoritmos:

Geração

Um algoritmo probabilístico $\mathcal{K}$ que gera um par de chaves pública e privada $(\text{ek}\|\text{dk})$.

Encapsulamento

Um algoritmo probabilístico $\mathcal{E}$ que recebe como input a chave pública ek e gera $(k\|\sigma)$ formado por um segredo k e o seu "encapsulamento" σ.

De-encapsulamento

Um algoritmo determinístico $\mathcal{D}$ que recebe $\sigma\|\text{dk}$ como input e gera k ou então falha.

- O **Data Encapsulation Mechanism (DEM)** é formado por 2 algoritmos:

Cifra

Um algoritmo probabilístico $\mathcal{E}_{\text{sym}}$ que recebe como input $k\|m$ (uma chave k e um texto m de comprimento arbitrário) e gera um criptograma y.

Decifra

Um algoritmo determinístico $\mathcal{D}_{\text{sym}}$ que recebe como input $k\|y$ (uma chave k e um criptograma de comprimento arbitrário y) e gera o texto m ou, então, falha.

A correcção do KEM exprime a propriedade de a chave k, gerada e encapsulada com a chave pública ek, ser a

mesma que se recupera do encapsulamento σ e da chave privada dk. Isto é

Para todo par de chaves $(\mathsf{ek}, \mathsf{dk})$ *, são indistinguíveis as variáveis aleatórias* k *e* k' *geradas pelos algoritmos*

$$k \| \sigma \leftarrow \mathcal{E}(\mathsf{ek}) \quad , \quad k' \leftarrow \mathcal{D}(\sigma \| \mathsf{dk})$$

De forma análoga a correcção do DEM exprime a capacidade de recuperar o texto m do criptograma y

Para toda a chave k *, são indistinguíveis as variáveis aleatórias* m *e* m' *determinada pelo algoritmo*

$$y \leftarrow \mathcal{E}_{\mathsf{sym}}(k \| m) \ ; \ m' \leftarrow \mathcal{D}_{\mathsf{sym}}(k \| y)$$

□

Obviamente é possível criar KEM´s usando identidades como chaves públicas. Por exemplo, o esquema IBE de Sakai-Kasahara (protocolo 42) pode ser adaptado a um esquema KEM.

Protocolo 47 : Esquema KEM orientado à identidade (inspirado no IBE de Sakai-Kasahara)

Setup & Extract os elementos comuns descritos no protocolo 40 e a geração de chaves do tipo II descrita no protocolo 41

Encapsulamento com a identidade $\mathcal{D}$, gera o segredo k e o seu encapsulamento σ.

(1) $\mathcal{E}: \upsilon \leftarrow \mathbb{B}^t \quad ; \quad k \leftarrow H(\upsilon)$ – *geração da chave*

(2) $\mathcal{E}: d \leftarrow \mathsf{ID}(\mathcal{D}) \quad ; \quad a \leftarrow H_r(\upsilon) \quad ; \quad \gamma \leftarrow (\beta\, g^d)^a$ – *encapsulamento*

(3) $\mathcal{E}: \mu \leftarrow \boldsymbol{\kappa}(g, g, a) \quad ; \quad \upsilon, \mu \rightarrow \sigma = \gamma \,\|\, (\upsilon \oplus f(\mu))$ – *emissão*

De-encapsulamento Com a chave privada λ e o encapsulamento $\sigma = \gamma \| \upsilon'$ recupera-se a chave k

(1) $\mathcal{D}: \gamma, \upsilon' \leftarrow \sigma \quad ; \quad \mu \leftarrow \boldsymbol{\kappa}(\gamma, \lambda, 1) \quad ; \quad \upsilon \leftarrow \upsilon' \oplus f(\mu)$ – *de-encapsulamento*

(2) $\mathcal{D}: a \leftarrow H_r(\upsilon) \quad ; \quad \gamma \stackrel{?}{=} (\beta\, g^d)^a$ – *confirmação*

(3) $\mathcal{D}: k \leftarrow H(\upsilon)$ – *recuperação da chave*

É possível construir variantes "escrow-free" destes esquemas IBE ou KEM-DEM usando duas formas de abordagem: com ou sem certificados.

Se tomar-mos como base o esquema de geração de chaves do tipo I, recorde-se que $\mathcal{D}$ tinha uma chave privada $\lambda = p^s$, sendo $p = \mathsf{ID}(\mathcal{D})$ e s a chave privada do TA. Aqui o TA conhece a chave privada de $\mathcal{D}$

Uma solução alternativa consiste em dar a $\mathcal{D}$ um segredo próprio x (não conhecido do TA) e usar, como informação privada λ, algo que dependa de x, para além da componente p^s conhecida pelo TA.

Isto sugere os dois seguintes esquemas KEM, "escrow-free", com e sem certificados.

No primeiro a função ID tem a aridade $\mathbb{G} \times \mathbb{B}^* \longrightarrow \mathbb{G}$; no segundo essa função tem a aridade usual $\mathbb{B}^* \longrightarrow \mathbb{G}$. No esquema sem certificados, estes são substituídos por um mecanismo de auto-certificação.

Protocolo 48 : KEM baseado em certificados (Gentry)

"Setup" TA gera a sua chave privada e publica a chave pública correspondente.

(1) $T: s \leftarrow \mathbb{Z}_r^* \quad ; \quad s \rightarrow \beta = g^s$

"Extract& Certify" $\mathcal{D}$ escolhe o segredo próprio e publicita a chave pública correspondente. TA certifica a chave pública ζ e emite a 2ª componente da chave privada de $\mathcal{D}$.

(1) $\mathcal{D}: x \leftarrow \mathbb{Z}_r^* \quad ; \quad x \rightarrow \zeta = g^x$

(2) $T: p \leftarrow \mathsf{ID}(\zeta, \mathcal{D}) \quad ; \quad z \leftarrow p^s$

(3) $\mathcal{D}: p \leftarrow \mathsf{ID}(\zeta, \mathcal{D}) \quad ; \quad \lambda \leftarrow (T.z) \cdot p^x$

Encapsulamento $\mathcal{E}$ gera a chave k e o seu encapsulamento σ.

(1) $\mathcal{E}: \upsilon \leftarrow \mathbb{B}^t \quad ; \quad k \leftarrow H(\upsilon)$ – *"keygen"*

(2) $\mathcal{E}: p \leftarrow \mathsf{ID}(\zeta, \mathcal{D}) \quad ; \quad a \leftarrow H_r(\upsilon) \quad ; \quad \gamma \leftarrow g^a$ – *redundância*

(3) $\mathcal{E}: \mu \leftarrow \boldsymbol{\kappa}(\beta, p, a) \cdot \boldsymbol{\kappa}(\zeta, p, a) \quad ; \quad \mu, \upsilon \rightarrow \sigma = \gamma \,\|\, \upsilon \oplus f(\mu)$ – *encapsulamento*

De-emcapsulamento $\mathcal{D}$ recupera k e verifica a sua autenticidade

(1) $\mathcal{D}: \gamma, y \leftarrow \sigma \quad ; \quad \mu \leftarrow \boldsymbol{\kappa}(\gamma, \lambda, 1) \quad ; \quad \upsilon \leftarrow y \oplus f(\mu) \quad ; \quad k \leftarrow H(\upsilon)$ – *extracção*

(2) $\mathcal{D}: a \leftarrow H_r(\upsilon) \quad ; \quad \gamma \stackrel{?}{=} g^a$ – *verificação*

Protocolo 49 : KEM sem certificados (Al-Riyami & Paterson)

"Setup" Como no protocolo 48

"Extract" $\mathcal{D}$ gera um segredo x, publica a respectiva informação pública ζ (que é auto-certificável) e extrai do TA a informação necessário ao cálculo da chave privada λ.

(1) $\mathcal{D}\colon x \leftarrow \mathbb{Z}_r^* \;\; ; \;\; x \rightarrow \zeta = g^x \parallel \beta^x$

(2) $T\colon p \leftarrow \mathrm{ID}(\mathcal{D}) \;\; ; \;\; z \leftarrow p^s$

(3) $\mathcal{D}\colon \lambda \leftarrow (T.z)^x$

Encapsulamento o emissor $\mathcal{E}$ gera a chave k e o seu encapsulamento σ

(1) $\mathcal{E}\colon q, t \leftarrow \zeta \;\; ; \;\; \mathrm{DDHP}(\beta, q, t) \stackrel{?}{=} 1$ – *certificação*

(2) $\mathcal{E}\colon \upsilon \leftarrow \mathbb{B}^t \;\; ; \;\; a \leftarrow H_r(\upsilon) \;\; ; \;\; \gamma \leftarrow g^a \;\; ; \;\; k \leftarrow H(\upsilon)$ – *redundância & "keygen"*

(3) $\mathcal{E}\colon p \leftarrow \mathrm{ID}(\mathcal{D}) \;\; ; \;\; \mu \leftarrow \boldsymbol{\kappa}(t, p, a) \;\; ; \;\; \upsilon, \mu \rightarrow \sigma = \gamma \parallel \upsilon \oplus f(\mu)$ – *encapsulamento*

De-encapsulamento $\mathcal{D}$ recupera k e verifica a sua autenticidade

(1) $\mathcal{D}\colon \gamma, y \leftarrow \sigma \;\; ; \;\; \mu \leftarrow \boldsymbol{\kappa}(\gamma, \lambda, 1) \;\; ; \;\; \upsilon \leftarrow y \oplus f(\mu) \;\; ; \;\; k \leftarrow H(\upsilon)$ – *extracção*

(2) $\mathcal{D}\colon a \leftarrow H_r(\upsilon) \;\; ; \;\; \gamma \stackrel{?}{=} g^a$ – *verificação*

No protocolo com certificados, a chave privada do agente $\mathcal{D}$ é $\lambda = p^{(s+x)}$. No de-emcapsulamento, $\mu = \boldsymbol{\kappa}(\gamma, \lambda, 1) = \boldsymbol{\kappa}(g^a, p^{(s+x)}, 1) = \boldsymbol{\kappa}(g, p, a\,(s+x))$. No encapsulamento tem-se $\mu = \boldsymbol{\kappa}(g^s, p, a) \cdot \boldsymbol{\kappa}(g^x, p, a) = \boldsymbol{\kappa}(g, p, (s+x)\,a)$.

No protocolo sem certificados, a chave privada é $\lambda = p^{s\,x}$. No encapsulamento $\mu = \boldsymbol{\kappa}(t, p, a) = \boldsymbol{\kappa}(g, p, s\,x\,a)$; no de-emcapsulamento $\mu = \boldsymbol{\kappa}(\gamma, \lambda, 1) = \boldsymbol{\kappa}(g^a, p^{s\,x}, 1) = \boldsymbol{\kappa}(g, p, s\,x\,a)$.

6.5 IBA sobre Emparelhamentos

Emparelhamentos e co-emparelhamentos tornam viáveis vários protocolos associados à autenticação: protocolos de identificação, esquemas de assinaturas, protocolos de acordo de chaves, etc. Tornam ainda viáveis protocolos mais complexos para assinatura e cifra simultânea de mensagens.

O mais simples destes protocolos será um protocolo de identificação que usa o esquema de geração de chaves do tipo I (ver pag. 396).

Protocolo 50 : Identificação (emparelhamentos)

"Setup&Extract" Protocolo de geração do tipo I (protocolo 38) em que um "prover" $\mathcal{P}$ recolhe a sua chave privada $\lambda = p^s$ com $p = \mathrm{ID}(\mathcal{P})$. A chave pública do TA é $\beta = g^s$.

Desafio O "verifier" $\mathcal{V}$ envia um desafio aleatório d

(1) $\mathcal{V}: \upsilon \leftarrow \mathbb{Z}_r^* \quad ; \quad \upsilon \rightarrow d = g^{\upsilon}$.

Resposta O "prover" $\mathcal{P}$ determina a resposta μ

(1) $\mathcal{P}: \lambda \rightarrow \mu = \boldsymbol{\kappa}(d, \lambda, 1)$

Verificação O "verifier" verifica a correcção da resposta usando o mecanismo $\boldsymbol{\kappa}$ (definição 131, pag. 388)

(1) $\mathcal{V}: p \leftarrow \mathrm{ID}(\mathcal{P}) \quad ; \quad \boldsymbol{\kappa}(\beta, p, \upsilon) \ =?= \ \mu$

A correcção do protocolo deriva das propriedades da função $\boldsymbol{\kappa}$

$$\mu \;=\; \boldsymbol{\kappa}(d,\lambda,1) \;=\; \boldsymbol{\kappa}(g^{\upsilon},p^{s},1) \;=\; \boldsymbol{\kappa}(g^{s},p^{1},\upsilon) \;=\; \boldsymbol{\kappa}(\beta,p,\upsilon)$$

O mesmo protocolo pode ser realizado com co-emparelhamentos. Para isso é necessário alterar ligeiramente os elementos comuns e a geração de chaves.

Assim assume-se que existe uma função co-emparelhamento $\mathbf{c}\colon \mathbb{G}\times\mathbb{G}'\to\mathbb{G}'$, sendo $\mathbb{G},\mathbb{G}'$ dois grupos de Diffie-Hellman com a mesma ordem prima r. A função de *hash* par determinação do representante da identidade é da forma $\mathsf{ID}\colon \mathbb{B}^*\to\mathbb{G}'\setminus 1$. Tal como na geração de chaves do tipo I, o TA tem uma chave privada $s\in\mathbb{Z}_r^*$ e uma chave pública $\beta=g^s$ e m agente I, com representante de identidade $p=\mathsf{ID}(I)$, tem chave privada $\lambda=p^s$.

Protocolo 51 : Identificação (co-emparelhamentos)

Desafio O "verifier" $\mathcal{V}$ envia um desafio aleatório d

(1) $\mathcal{V}\colon \upsilon\leftarrow\mathbb{Z}_r^* \quad ; \quad \upsilon\to d=g^{\upsilon}.$

Resposta O "prover" $\mathcal{P}$ determina a resposta μ usando o co-emparelhamento

(1) $\mathcal{P}\colon \lambda\to\mu=\mathbf{c}(d,\lambda)$

Verificação O "verifier" verifica a correcção da resposta usando também o co-emparelhamento

(1) $\mathcal{V}\colon p\leftarrow\mathsf{ID}(\mathcal{P}) \quad ; \quad \mathbf{c}(\beta,p^{\upsilon}) \;=?=\; \mu$

A correcção do protocolo deriva da propriedade bilenear da função de co-emparelhamento: $\mathbf{c}(g^{\upsilon}, p^{s}) = p^{\upsilon\, s} = \mathbf{c}(g^{s}, p^{\upsilon})$.

□

Usando emparelhamentos (algoritmo 3) e co-emparelhamentos (algoritmo 4) é possível construir oráculos DDHP e, dessa forma, implementar assinaturas e protocolos de identificação que requeiram apenas a estrutura de um grupo Gap Diffie-Hellman. Estão nesta categoria as assinatura de Cha & Cheon (protocolo 32) e BLS (protocolo 34) e o esquema de identificação de Cha & Cheon (protocolo 33).

Outras tecnicas de autenticação requerem as proptiedades algébricas dos emparelhamentos. Um exemplo paradigmático é o esquema de assinatura de Hess baseado no "setup" e geração de chaves do tipo Boneh & Franklin.

Protocolo 52 : Esquema de assinaturas IBA de Hess

"Setup&Extract" Usa os elementos comuns e a geração de chaves do tipo I (protocolo 38). Usa também uma função de *hash* $H' : \Gamma \times \mathbb{B}^t \to \mathbb{Z}_r$.

Assinatura O "prover" $\mathcal{P}$ constrói uma assinatura σ para a mensagem $m \in \mathbb{B}^t$

(1) $\mathcal{P} : \upsilon \leftarrow \mathbb{Z}_r^* \quad ; \quad \mu \leftarrow \boldsymbol{\kappa}(\lambda, g, \upsilon) \quad ; \quad h \leftarrow H'(\mu, m) \quad ; \quad \upsilon, \lambda, h \to \sigma = h \| \lambda^{\upsilon - h}$

Verificação com a chave pública β do TA e a identidade do "prover", verifica a assinatura σ no texto m.

(1) $\mathcal{V} : h, \alpha \leftarrow \sigma \quad ; \quad p \leftarrow \mathsf{ID}(\mathcal{P}) \quad ; \quad \mu' \leftarrow \boldsymbol{\kappa}(\alpha, g, 1) \cdot \boldsymbol{\kappa}(p, \beta, h)$

(2) $\mathcal{V} : h \ =?= \ H'(\mu', m)$

Notas

1. A função H' pode ser gerada a partir dos elementos já existentes; por exemplo pode-se definir $H'(\mu, m) = H_r(f(\mu) \oplus m)$.
2. A correcção deriva das propriedades algébricas do emparelhamento.
   Temos $\lambda = p^s$ e $\beta = g^s$; na verificação tem-se $\alpha = \lambda^{\upsilon - h} = p^{s\,(\upsilon - h)}$; se representarmos por γ o valor $\boldsymbol{\kappa}(p, g, 1)$, tem-se
   $$\mu = \boldsymbol{\kappa}(\lambda, g, \upsilon) = \gamma^{s\,\upsilon} \qquad , \qquad \mu' = \boldsymbol{\kappa}(\alpha, g, 1) \cdot \boldsymbol{\kappa}(p, \beta, h) = \gamma^{s\,(\upsilon - h)} \cdot \gamma^{s\,h}$$
   Consequentemente o valor de μ calculado na assinatura coincide com o valor de μ' calculado na verificação.
3. Este esquema pode ser imediatamente adaptado a co-emparelhamentos $\mathbf{c} \colon \mathbb{G} \times \Gamma \to \Gamma$. Para isso a função de *hash* ID deve dar resultados em Γ e μ tem também um valor neste segundo grupo.
   Na assinatura, μ calcula-se como $\mu \leftarrow \mathbf{c}(g^{\upsilon}, \lambda)$; na verificação μ calcula-se como $\mu \leftarrow \mathbf{c}(g, \alpha) \cdot \mathbf{c}(\beta, p^{h})$.
   Em tudo o resto o esquema mantém-se inalterado e é fácil ver que está correcto.

Uma variante do esquema de assinaturas de Cha & Cheon (protocolo 32) pode ser construída usando um "setup" e geração de chaves do tipo II (protocolo 41) e designa-se por **assinaturas BLQM**.

O esquema BLQM aqui apresentado usa emparelhamentos e a função $\boldsymbol{\kappa}$. Este esquema pode ser transformado num esquema análogo que use co-emparelhamentos $\mathbf{c} \colon \mathbb{G} \times \Gamma \to \Gamma$. Recomenda-se ao aluno que, como exercício, construa uma versão do BLQM usando co-emparelhamentos.[51].

[51] Sugestão: ver a adaptação do esquema IBE de Sakay & Kasahara aos co-emparelhamentos

Protocolo 53 : Esquema de Assinaturas BLMQ (Barreto, Libert, McCullagh & Quisquater)

"Setup&Extract" Usa os elementos comuns e a geração de chaves do tipo II (protocolo 41,pag. 400). A função de *hash* H_r tem a aridade $H_r : \Gamma \times \mathbb{B}^* \to \mathbb{Z}_r^*$.

Assinatura O "prover" $\mathcal{P}$ assina o texto m.

(1) $\mathcal{P} : \upsilon \leftarrow \mathbb{Z}_r^* \quad ; \quad \mu \leftarrow \boldsymbol{\kappa}(g, g, \upsilon) \quad ; \quad h \leftarrow H_r(\mu, m)$

(2) $\mathcal{P} : \lambda, \upsilon \to \sigma = h \| \lambda^{\upsilon + h}$

Verificação O "verifier" $\mathcal{V}$ verifica a assinatura σ de $\mathcal{P}$ no texto m usando chave pública β do TA.

(1) $\mathcal{V} : h, \alpha \leftarrow \sigma \quad ; \quad p \leftarrow \mathrm{ID}(\mathcal{P}) \quad ; \quad \mu' \leftarrow \boldsymbol{\kappa}(g, g, -h) \cdot \boldsymbol{\kappa}(\alpha, \beta\, g^p, 1)$

(2) $\mathcal{V} : h \ =?= \ H_r(\mu', m)$

A correcção deriva do facto de ser $\lambda = g^{(s+p)^{-1}}$ e $\beta = g^s$. Donde, o termo $\beta\, g^p$ tem o valor $g^{(s+p)}$ e $\alpha = g^{(s+p)^{-1} \cdot (\upsilon + h)}$. Se representar-mos por γ o valor $\boldsymbol{\kappa}(g, g, 1)$, tem-se

$$\mu = \gamma^{\upsilon} \quad , \quad \mu' = \gamma^{-h} \cdot \gamma^{(s+p)^{-1} \cdot (\upsilon + h) \cdot (s+p)} = \gamma^{-h} \cdot \gamma^{(\upsilon + h)} = \gamma^{\upsilon}$$

Como $\mu = \mu'$, tem-se $h = H_r(\mu', m)$.

6.6 Autenticação mútua em IBC

Nas secções anteriores vimos esquemas e protocolos onde, essencialmente, só um dos agentes (para além do TA) tinha necessidade de chave privada porque só esse agente necessitava de ser autenticado.

Nesta secção vamos ver técnicas criptográficas que envolvem a autenticação mútua de dois agentes distintos A e B e, portanto, ambos os agentes necessitam de chaves privadas. Estão nesta categoria os protocolos de acordo de chaves e os protocolos "sygn-encrypt" que assinam e cifram, simultaneamente, uma mensagem.

O exemplo mais simples de protocolo que requer duas chaves privadas é o protocolo de Diffie-Hellman (protocolo 6, pág. 303). Resumidamente

> *Num grupo DH de gerador* g*, o agente* A*, com chave privada* a*, torna público* g^a *e o agente* B*, com chave privada* b*, torna público* g^b*.*

É bem conhecido que este protocolo simples de está sujeito ao ataque de "homem-no-meio" porque não tem autenticação mútua dos agentes intervenientes.

O protocolo tem, no entanto, a vantagem de trocar mensagens muito simples: os valores g^a e g^b. Será interressante manter estas mensagens (ou semelhantes) mas acrescentar a autenticação dos agentes. Isso torna-se possível se recorrer-mos a emparelhamentos ou co-emparelhamentos.

Protocolo 54 : Acordo de Chaves (Smart, Chen & Kudla)

"Setup" & "Extract´´" Tipo I, sendo s a chave privada do TA, $\beta = g^s$ a sua chave pública. As chaves privadas dos agentes intervenientes são $\lambda_A = q^s$ e $\lambda_B = p^s$ sendo $q = \mathsf{ID}(A)$ e $p = \mathsf{ID}(B)$.

"Run" Troca de mensagens

(1) $A\colon a \leftarrow \mathbb{Z}_r^* \quad ; \quad a \rightarrow \sigma_A = g^a$

(2) $B\colon b \leftarrow \mathbb{Z}_r^* \quad ; \quad b \rightarrow \sigma_B = g^b$

Construção Os agentes constroem a mesma chave e autenticam-se mutuamente

(1) $A\colon p \leftarrow \mathsf{ID}(B) \quad ; \quad \mu_A \leftarrow \boldsymbol{\kappa}(\lambda_A, \sigma_B, 1) \cdot \boldsymbol{\kappa}(p, \beta, a) \quad ; \quad \kappa_A \leftarrow (\sigma_B)^a \,\|\mu_A$

(2) $B\colon q \leftarrow \mathsf{ID}(A) \quad ; \quad \mu_B \leftarrow \boldsymbol{\kappa}(\lambda_B, \sigma_A, 1) \cdot \boldsymbol{\kappa}(q, \beta, b) \quad ; \quad \kappa_B \leftarrow (\sigma_A)^b \,\|\mu_B$

Notas

1. Porque $\lambda_A = q^s$ e $\lambda_B = p^s$, tem-se

$$\mu_A = \boldsymbol{\kappa}(q, g, s\,b) \cdot \boldsymbol{\kappa}(p, g, s\,a) \qquad \text{e} \qquad \mu_B = \boldsymbol{\kappa}(p, g, s\,a) \cdot \boldsymbol{\kappa}(q, g, s\,b)$$

   Portanto $\mu_A = \mu_B$; como $(\sigma_B)^a = g^{a\,b} = (\sigma_A)^b$, tem-se $\kappa_A = \kappa_B$; i.e. ambos os agentes reconstroem a mesma chave.
2. Existe autenticação mútua dos agentes porque a chave κ construída por cada um desses agentes depende da identidade pública do outro agente e da chave privada do agente que a constrói. Um eventual intruso C não consegue passar informação a A que o faça calcular uma chave pré-definida por C pensando que está a interagir com B.

A informação de autenticação está na componente $\mu_A = \mu_B$ da chave. Note-se que essa componente é calculada como o produto de dois termos em que cada um deles contém uma parte de informação pública (a chave pública β ou uma das mensagens) e uma parte de informação privada (a chave privada ou o segredo gerado localmente). Esta observação faz com que, escolhendo outros termos com estas caracteristicas, seja possível gerar variantes deste esquema básico:

Variante com co-emparelhamentos

Existe uma simples adaptação deste protocolo a co-emparelhamentos $\mathbf{c}\colon \mathbb{G} \times \Gamma \to \Gamma$. Para isso a função de *hash* tem de dar valores em Γ e constói-se

$$\mu_A \leftarrow \mathbf{c}(\sigma_B, \lambda_A) \cdot \mathbf{c}(\beta, p^a) \quad , \quad \mu_B \leftarrow \mathbf{c}(\sigma_A, \lambda_B) \cdot \mathbf{c}(\beta, q^b)$$

Tem-se então $\mu_A = \mu_B = \mathbf{c}(g, q)^{s\,b} \cdot \mathbf{c}(g, p)^{s\,a}$.

Variante Chen&Kudla

Neste caso as mensagens trocadas são

$$A\colon a \to \sigma_A = q^a \quad , \quad B\colon b \to \sigma_B = p^b$$

As chaves são calculadas como

$$\mu_A \leftarrow \boldsymbol{\kappa}(\lambda_A, \sigma_B \cdot p^a, 1) \quad , \quad \mu_B \leftarrow \boldsymbol{\kappa}(\sigma_A \cdot q^b, \lambda_B, 1)$$

É simples verificar que $\mu_A = \mu_B = \boldsymbol{\kappa}(q, p, s\,(a+b))$.

Variante Chen&Kudla com emparelhamentos

Exercício a cargo do aluno.

Um esquema similar pode ser construído com geração de chaves do tipo II.

Protocolo 55 : Acordo de Chaves (McCullagh & Barreto)

"Setup" & "Extract" Tipo II, sendo s a chave privada do TA, $\beta = g^s$ a sua chave pública. As chaves privadas dos agentes intervenientes são $\lambda_A = g^{f(s,q)}$ e $\lambda_B = g^{f(s,p)}$ sendo $q = \mathrm{ID}(A)$ e $p = \mathrm{ID}(B)$ e $f(s,x) = (s+x)^{-1}$.

"Run" Troca de mensagens

(1) $A\colon a \leftarrow \mathbb{Z}_r^* \;;\; p \leftarrow \mathrm{ID}(B) \;;\; a \rightarrow \sigma_A = (g^p \cdot \beta)^a$

(2) $B\colon b \leftarrow \mathbb{Z}_r^* \;;\; q \leftarrow \mathrm{ID}(A) \;;\; b \rightarrow \sigma_B = (g^q \cdot \beta)^b$

Construção Os agentes constroem a mesma chave e autenticam-se mutuamente

(1) $A\colon \kappa_A \leftarrow \boldsymbol{\kappa}(\lambda_A, \sigma_B, 1) \cdot \boldsymbol{\kappa}(g, g, a)$

(2) $B\colon \kappa_B \leftarrow \boldsymbol{\kappa}(\lambda_B, \sigma_A, 1) \cdot \boldsymbol{\kappa}(g, g, b)$

Note-se que $\sigma_B = g^{b\,(s+q)}$. Atendendo à forma de λ_A tem-se $\kappa_A = \boldsymbol{\kappa}(g, g, f(s,q) \cdot b \cdot (s+q)) \cdot \boldsymbol{\kappa}(g, g, a) = \boldsymbol{\kappa}(g, g, a+b)$. Dualmente se mostra que $\kappa_B = \boldsymbol{\kappa}(g, g, a+b) = \kappa_A$.

Este protocolo também se adapta facilmente ao uso de co-emparelhamentos. Como exercício o aluno deve construir tal adaptação.

□

As técnicas IBC mais interessantes, envolvendo autenticação mútua de agentes, são os **protocolos "sign-encryption"** (abreviadamente **"signcryption"**) que combinam esquemas de assinaturas com esquemas de cifra.

Nestes protocolos, o emissor $\mathcal{E}$ cifra uma mensagem m com a identidade do agente $\mathcal{D}$ como chave pública e, simultaneamente, assina essa mensagem com a sua chave privada. Por seu lado, $\mathcal{D}$ verifica a autenticidade da mensagem usando a identidade de $\mathcal{E}$ como chave pública e recupera m do criptograma, com a sua chave privada.

Isto significa que ambos os agentes têm de ter chaves privadas: $\mathcal{E}$ precisa de uma chave privada para assinar a mensagem e $\mathcal{D}$ precisa de uma chave privada para decifrar a mensagem.

Vamos apresentar dois protocolos de "signcryption" baseados em emparelhamentos: um deles usa geração de chaves do tipo I e o outro usa geração de chaves do tipo II. Ambos podem ser facilmente adpatados a co-emparelhamentos ficando a cargo do aluno realizar as modificações apropriadas.

Protocolo 56 : Signcryption (Chen, Malone & Lee)

"Setup" & "Extract´´" Tipo I, sendo s a chave privada do TA, $\beta = g^s$ a sua chave pública. As chaves privadas dos agentes intervenientes são $\lambda_e = e^s$ e $\lambda_d = d^s$ sendo $e = \mathrm{ID}(\mathcal{E})$ e $d = \mathrm{ID}(\mathcal{D})$.
São usadas funções de*hash* $f : \Gamma \to \mathbb{B}^t$ e $H_r : \mathbb{G} \times \mathbb{B}^t \to \mathbb{Z}_r^*$.

"Sign-Encrypt" $\mathcal{E}$ assina a mensagem $m \in \mathbb{B}^t$ destinada a $\mathcal{D}$ e cifra-a; produz um "signed-cryptogram" σ.

(1) $\mathcal{E} : \upsilon \leftarrow \mathbb{Z}_r^* \;;\; \gamma \leftarrow e^{\upsilon} \;;\; h \leftarrow H_r(\gamma, m) \;;\; \kappa \leftarrow \lambda_e^{h+\upsilon} \,\|\, \mathcal{E} \,\|\, m$ – *sign*

(2) $\mathcal{E} : \mu \leftarrow \boldsymbol{\kappa}(\lambda_e, d, \upsilon) \;;\; \mu, \kappa, \gamma \to \sigma = \gamma \,\|\, (f(\mu) \oplus \kappa)$ – *encrypt*

"Decrypt-Verify" $\mathcal{D}$ recupera a mensagem e verifica a sua autenticidade.

(1) $\mathcal{D} : \gamma, y \leftarrow \sigma \;;\; \mu \leftarrow \boldsymbol{\kappa}(\gamma, \lambda_d, 1) \;;\; \alpha, \mathcal{E}, m \leftarrow f(\mu) \oplus y$ – *decrypt*

(2) $\mathcal{D} : e \leftarrow \mathrm{ID}(\mathcal{E}) \;;\; h \leftarrow H_r(\gamma, m) \;;\; \mathrm{DDHP}(\beta, \gamma \cdot e^h, \alpha) \;=?=\; 1$ – *verify*

1. O emissor $\mathcal{E}$ começa por substituir a mensagem m por uma versão κ autenticada da mesma, juntando-lhe a sua própria identidade $\mathcal{E}$ e uma assinatura $\alpha = \lambda_e^{(\upsilon+h)}$; κ é depois cifrada a partir de uma chave μ convertida para o espaço apropriado pela função $f(\cdot)$. A redundância γ liga estes dois passos e vai permitir recuperar a chave μ e verificar a assinatura.
2. Na operação de cifra, a "chave" μ é calculada como $\boldsymbol{\kappa}(\lambda_e, d, \upsilon) = \boldsymbol{\kappa}(e, d, s\,\upsilon)$. Na operação de decifra é calculada como $\boldsymbol{\kappa}(\gamma, \lambda_d, 1) = \boldsymbol{\kappa}(e^{\upsilon}, d^s, 1) = \boldsymbol{\kappa}(e, d, \upsilon\, s)$. A chave é a mesma e $\mathcal{D}$ consegue recuperar, para além da mensagem m, a assinatura α e a identidade do emissor $\mathcal{E}$.
3. A verificação da assinatura recorre a um oráculo DDHP (que pode ser implementado com a mesma função $\boldsymbol{\kappa}$), testando se o triplo formadado por $\beta = g^s$, pelo termo $\gamma \cdot e^h = e^{(\upsilon+h)}$ e $\alpha = \lambda_e^{(\upsilon+h)} = e^{s\,(\upsilon+h)}$ formam um triplo DH.

Em seguida mostra-se uma versão modificada do protocolo BLMQ em que, tal como no esquema anterior, a assinatura de m precede a cifra.

Protocolo 57 : BLMQ Signcryption Scheme (versão "first sign")

"Setup" & "Extract" Tipo II, sendo s a chave privada do TA, $\beta = g^s$ a sua chave pública. As chaves privadas do emissor e destinatário são $\lambda_e = g^{t(s,e)}$ e $\lambda_d = g^{t(s,d)}$ sendo $e = \mathsf{ID}(\mathcal{E})$ e $d = \mathsf{ID}(\mathcal{D})$ e $t(s,x) = (s+x)^{-1}$.

São usadas funções de *hash* $f : \Gamma \to \mathbb{B}^t$ e $H_r : \Gamma \times \mathbb{B}^t \to \mathbb{Z}_r^*$.

"Sign-Encrypt" $\mathcal{E}$ cifra a mensagem $m \in \mathbb{B}^t$ destinada a $\mathcal{D}$ e assina-a; produz um "signed-cryptogram" σ.

(1) $\mathcal{E} : \upsilon \leftarrow \mathbb{Z}_r^* \;;\; \mu \leftarrow \boldsymbol{\kappa}(g,g,\upsilon) \;;\; h \leftarrow H_r(\mu, m) \;;\; \kappa \leftarrow \lambda_e^{(\upsilon+h)} \,\|\, \mathcal{E} \,\|\, m$ *– sign*

(2) $\mathcal{E} : d \leftarrow \mathsf{ID}(\mathcal{D}) \;;\; \gamma \leftarrow g^{d\,\upsilon} \cdot \beta^{\upsilon} \;;\; \gamma, \mu, \kappa \to \sigma = \gamma \,\|\, (f(\mu) \oplus \kappa)$ *– encrypt*

"Decrypt-Verify" $\mathcal{D}$ recupera a mensagem e verifica a sua autenticidade.

(1) $\mathcal{D} : \gamma, y \leftarrow \sigma \;;\; \mu \leftarrow \boldsymbol{\kappa}(\gamma, \lambda_d, 1) \;;\; \alpha, \mathcal{E}, m \leftarrow f(\mu) \oplus y$ *– decrypt*

(2) $\mathcal{D} : h \leftarrow H_r(\mu, m) \;;\; e \leftarrow \mathsf{ID}(\mathcal{E}) \;;\; \mu \cdot \boldsymbol{\kappa}(g,g,h) \stackrel{?}{=} \boldsymbol{\kappa}(\alpha, g^e \cdot \beta, 1)$ *– verify*

1. Tal como no protocolo 56, o emissor substitui a mensagem m por uma versão autenticada κ que contém, para além de m, a assinatura

$\alpha = \lambda_e^{(\upsilon+h)}$ e a sua identidade $\mathcal{E}$. A ligação entre a assinatura e a cifra é feita por uma chave comum μ usada tanto para construir o *hash* $h = Hr(\mu, m)$ como a chave de cifra $f(\mu)$. Na redundância γ vai estar incluída a identidade $\mathcal{D}$ do destinatário de forma a este poder, com a sua chave privada, recuperar μ.

2. Note-se que $\gamma = g^{\upsilon\,(d+s)}$ e $\lambda_d = g^{t(s,d)}$ com $t(s,d) = (d+s)^{-1}$; portanto

$$\boldsymbol{\kappa}(\gamma, \lambda_d, 1) \;=\; \boldsymbol{\kappa}(g, g, \upsilon \cdot (d+s) \cdot (d+s)^{-1}) \;=\; \boldsymbol{\kappa}(g, g, \upsilon) \;=\; \mu$$

Para verificar a assinatura note-se que

$$\mu \cdot \boldsymbol{\kappa}(g, g, h) \;=\; \boldsymbol{\kappa}(g, g, \upsilon) \cdot \boldsymbol{\kappa}(g, g, h) \;=\; \boldsymbol{\kappa}(g, g, \upsilon + h)$$

Como $\alpha = \lambda_e^{(\upsilon+h)}$, sendo $\lambda_e = g^{t(s,e)}$ e $t(s,e) = (s+e)^{-1}$, tem-se também

$$\boldsymbol{\kappa}(\alpha, g^e \cdot \beta, 1) \;=\; \boldsymbol{\kappa}(\alpha, g^{(s+e)}, 1) \;=\; \boldsymbol{\kappa}(g, g, (s+e)^{-1}\,(\upsilon+h)\,(s+e)) \;=\; \boldsymbol{\kappa}(g, g, \upsilon + h)$$

Os dois esquemas "signcryption" são construídos de forma à identidade do emissor estar protegida no criptograma: o destinatário precisa de decifrar a mensagem para saber qual foi o emissor. Desta forma estes esquemas não só garantem confidencialidade da mensagem como também a privacidade dos intervenientes.

7.Criptografia com Agentes Múltiplos

Frequentemente um determinado passo de uma técnica criptográfica pode requerer a participação de mais do que um agente. Suponhamos, por exemplo, que duas instituições pretendem estabelecer um contracto entre si, contracto esse que tem de ser assinado digitalmente.

Nos esquemas de assinaturas que conhecemos, o acto de assinar é realizado por indivíduos (e não instituições) que são titulares de chaves privadas próprias; não faz qualquer sentido propor-se, por exemplo, uma chave privada institucional porque, por definição, não é privada. Como as chaves privadas são individuais temos de ter um mecanismo de assinaturas que permita a vários indivíduos assinar em nome da instituição.

Pode ser exigida, adicionalmente, mais do que uma assinatura individual para comprometer a instituição; por exemplo, se a instituição tiver 3 directores, pode-se exigir a assinatura individual de quaisquer dois deles para realizar a assinatura institucional. Quando são necessárias várias assinaturas individuais para gerar uma assinatura institucional, também se pode exigir que elas sejam realizadas por uma determinada ordem.

Esboçamos uma instância daquilo que designaremos por **multi-assinatura**. Basicamente uma multi-assinatura é um acto (a assinatura institucional) que assume duas formas principais:

Multi-assinaturas "treshold" (t, n) A assinatura é uma assinatura individual que recorre a informação privada que está distribuída por n agentes; o concluio entre um número de agentes $\leq t$ não é suficiente para determinar essa informação privada; é necessário combinar as chaves privadas de, pelo menos, $t + 1$ desses agentes para construir a chave privada.

Multi-assinaturas estruturadas O esquema de assinaturas é um algoritmo que usa, como oráculo, a realização de assinaturas individuais de vários agentes segundo uma ordem pré-estabelica. Normalmente o acto institucional pode ser realizado por mais do que uma dessas *cadeias de assinaturas*.

Uma outra abordagem, designada por **assinatura de grupo**, define grupos de assinantes individuais que podem contribuir para determinar a assinatura institucional; qualquer assinatura individual dentro do grupo realiza a assinatura institucional; isto corresponde a uma multi-assinatura onde todas as cadeias têm comprimento 1. No entanto exige-se algo mais: o *anonimato*; exige-se que a verificação da assinatura institucional reconheça a autenticidade do documento e a sua autoria (em termos da instituição emissora) mas não seja capaz de identificar a o titular da assinatura individual que lhe deu origem.

□

Nos esquemas de cifra podem-se definir procesos duais. Por simplicidade vamos apenas referir a mecanismos KEM-DEM já que cifras assimétricas são directamente implementáveis com (ou deriváveis de) este tipo de mecanismos.

Um esquema de **multi-KEM** envolve o agente *emissor* $\mathcal{E}$ e vários *destinatários* $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_n$. O emissor gera

uma chave k e um encapsulamento σ recorrendo às chaves públicas $p_1, p_2, \cdots, p_n$ dos vários destinatários de tal forma que, com qualquer uma das chaves privadas correspondentes $s_1, s_2, \cdots, s_n$, é possível recuperar k a partir do encapsulamento.

Resumidamente, o encapsulamento é um algoritmo probabilístico que produz

$$k \,\|\, \sigma \quad \leftarrow \quad \mathsf{mKEM}(p_1, p_2, \cdots, p_n)$$

e o desencapsulamento é um algoritmo determinístico que produz

$$k \quad \leftarrow \quad \mathsf{mKEM}(s_i, \sigma)^{-1} \qquad \forall\, i \in 1..n$$

No esquema multi-KEM qualquer titular de uma das chaves privadas s_i pode reconstruir a chave k e, assim, decifrar qualquer mensagem que tenha sido cifrada com ela. O objectivo é ransmitir a mesma mensagem a múltiplos destinatários reutilizando o mesmo encapsulamento; em relação a um esquema KEM singular, usado uma vez para cada destinatário, poupa-se $(n-1)$ encapsulamentos.

Outro tipo de objectivo envolve apenas um emissor $\mathcal{E}$ e um destinatário $\mathcal{D}$ mas exige que a chave privada usada recuperar k, seja construída a partir de várias fracções. Como caso paricular do de-emcapsulamento multi-chave é usado no chamado **"workflow decryption"**.

Neste processo decifrar é uma operação privilegiada que exige, ao agente destinatário do criptograma, o acesso a um

conjunto de *credenciais* fornecidas por ou mais *autoridades de credenciamento*. A interacção entre os vários agentes rege-se pelos seguintes princípios:

1. O **emissor** especifica o conjunto de credenciais necessárias para decifrar a mensagem numa *"policy"* determinada antes de cifrar a mensagem. A cifra é realizada independentemente das credenciais que estão ou não estão emitidas.
2. As **autoridades de credenciamento** emitem as credenciais no pressuposto que ocorreram determinados eventos. Desta forma as autoridades podem controlar a sequência de eventos (o "workflow") que conduz a um objectivo final de decifrar o criptograma.
3. Cumprindo os requisitos das autoridades, o **destinatário** vai recolhendo as credenciais e, com elas, construindo a chave que permite decifrar o criptograma. Cada credencial age como uma parte da chave privada.
4. Emissor e destinatário podem querer que o protocolo seja "escrow-free"; isto é, por si só, as autoridades de credenciamento não conseguem reunir informação que lhes permite, em conjunto, reconstruir a chave privada.

7.1 Esquemas de Partição de Chaves

O esquema de assinaturas BLS (protocolo 34 na página 384) pode ser facilmente generalizável a vários tipos de objectivos. Vimos que podia ser adaptada a um esquema de assinaturas cegas (protocolo 35, pag. 385) e iremos ver agora como se pode adaptar aos objectivos das multi-assinaturas.

Recordemos que no protocolo BLS básico, "prover" $\mathcal{P}$ detém uma chave privada s e publica a chave pública $\beta = g^s$. Se for h o *hash* da mensagem, $\mathcal{P}$ constrói a assinatura como $\sigma = h^s$.

Uma primeira abordagem consiste em usar um esquema de distribuição do segredo s por vários agentes de uma forma análoga à usada no protocolo 39 (pag. 397) para distribuir por vários agentes a responsabilidade de geração de uma chave privada.

Assume-se que existem vários "provers" $\mathcal{P}_1, \cdots, \mathcal{P}_n$ que são titulares de chaves privadas $s_1, \cdots, s_n \in \mathbb{Z}_r^*$ a que correspondem chaves públicas $\beta_1, \cdots, \beta_n \in \mathbb{G}$ sendo $\beta_i = g^{s_i}$.

Para cada "prover" $\mathcal{P}_i$, tal como no esquema BLS, sobre um determinado *hash* h o segredo s_i determina a assinatura $\sigma_i = h^{s_i}$; o triplo $\langle \beta_i, h, \sigma_i \rangle$ é um triplo DH.

134 Definição

*Uma n-**máscara** em $\mathbb{Z}_r$ é um vector $\mathbf{b} = (b_1, b_2, \ldots, b_n) \in (\mathbb{Z}_r)^n$ de n componentes em $\mathbb{Z}_r$. A máscara é **booleana** se todas as componentes forem 0 ou 1. O **suporte** da máscara é o sub-conjunto de índices i para os quais $b_i \neq 0$; a cardinalidade deste conjunto chama-se **tamanho** da máscara e representa-se por $|\mathbf{b}|$.*

Cada máscara $\mathbf{b}$ permite construir um segredo global s, uma chave pública global β e uma assinatura global σ do modo usual:

$$s = \sum_{i=1}^{n} s_i \cdot b_i \quad , \quad \beta = \prod_{i=1}^{n} (\beta_i)^{b_i} \quad , \quad \sigma = \prod_{i=1}^{n} (\sigma_i)^{b_i} \tag{120}$$

É fácil verificar que g^s coincide com β e que h^s coincide com σ; assim $\langle \beta, h, \sigma \rangle$ é também um triplo DH.

Conforme a estratégia para determinação da máscara $\mathbf{b}$ e, por conseguinte, construção de σ e β, definem-se variantes do esquema básico BLS envolvendo múltiplos assinantes. Essencialmente existe uma escolha entre:

Não coordenação da chave pública

Os "provers" $\mathcal{P}_i$ não coordenam a sua acção: cada assinante gera a publica, independentemente dos restantes, a chave pública β_i e, para cada *hash* h, a assinatura σ_i.

Cabe ao "verifier" escolher os assinantes em quem confia; faz isso gerando uma máscara boolena $\mathbf{b}$ booleana, cujo suporte coincide com esse conjunto de assinantes, e com ela calcular tanto β como σ.

Coordenação na chave pública

Existe um agente coordenador que distribui as chaves privadas s_i pelos "provers" e é o único que conhece o segredo global s e, por isso, é o único capaz de gerar a chave pública $\beta = g^s$. A verificação da assinatura recorre sempre a esta chave pública única.

Na hipótese de não-coordenação da chave pública o esquema BLS com múltiplos "provers" pode ser:

Protocolo 58 : Esquemas de Assinaturas BLS com n-máscaras

"Setup" Os "provers" $\mathcal{P}_1, \cdots, \mathcal{P}_n$ geram as chaves privadas e publicam as chaves públicas correspondentes.

(1) $\mathcal{P}_i$: $s_i \leftarrow \mathbb{Z}_r^*$; $s_i \rightarrow \beta_i = g^{s_i}$ para $i = 1..n$

Assinatura Sobre o mesma mensagem M cada um dos "provers" gera uma assinatura

(1) $\mathcal{P}_i$: $h \leftarrow H(M)$; $s_i \rightarrow \sigma_i = h^{s_i}$ para $i = 1..n$

Construção Usando uma máscara $\mathbf{b}$, o "verifier" constrói a assinatura global σ e a chave pública global β.

(1) $\mathcal{V}$: $\sigma \leftarrow \prod_i (\sigma_i)^{b_i}$; $\beta \leftarrow \prod_i (\beta_i)^{b_i}$

Verificação $\mathcal{V}$ verifica a assinatura σ do texto M com a chave pública β

(1) $\mathcal{V}$: $h \leftarrow H(M)$; $\mathrm{DDHP}(\beta, h, \sigma)$

Na hipótese de existir uma chave pública única β, a geração da assinatura σ recorre a duas estratégias possíveis:

Geração pelo verificador / assinatura não-anónima

A geração da assinatura não é coordenada; um sub-conjunto dos "provers" gera e publica assinaturas individuais σ_i. Cabe ao verificador recolher estas assinaturas, recolher a identidade dos "provers" assinantes e, se tiver informação suficiente, reconstruir a assinatura global σ.

Geração por um "prover" privilegiado / assinatura anónima

Existe um "prover" coordenador que recolhe um número suficiente de assinaturas individuais σ_i e a identidade dos votantes respectivos. Com esta informação constrói a assinatura global que publica.

A diferença crucial entre estas duas estratégias reside no facto de, na primeira construção, a identidade dos assinantes é conhecida pelo verificador enquanto que na segunda opção essa identidade está escondida do verificador. Assim a primeira assinatura não é anónima enquanto a segunda é completamente anónima.

□

Em qualquer dos casos, o facto de o segredo s ser "distribuído" pelos vários "provers" $\mathcal{P}_i$ leva-nos a um dos problemas clássicos em Criptografia, o **problema da partição de segredos**. Essencialmente este problema pode-se definir do seguinte modo:

Problema da Partição de Segredos

Dados inteiros positivos $t \leq n$ *e um domínio recursivo enumerável* D, *gerar um segredo* $\mathbf{s} \in D$ *e um conjunto* $\mathcal{S} = \{z_1, z_2, \ldots, z_n\}$ *de items chamados* **sombras** *ou* **quotas**, *de tal forma que:*

1. *O conhecimento de qualquer sub-conjunto* $R \subseteq \mathcal{S}$, *de cardinalidade inferior a um limite* t, *não permite conhecer* $\mathbf{s}$,
2. *Existe um algoritmo PPT, designado* **algoritmo de recuperação**, *que, sob input de um qualquer* $R \subseteq \mathcal{S}$ *falha se* $|R| < t$ *e dá o resultado* $\mathbf{s}$, *se* $|R| \geq t$.

Na terminologia inglesa uma solução para este problema designa-se por (t, n) **threshold scheme**. Este problema tem sido amplamente estudado desde que foi introduzido por Shamir & Blakley em 1979.

Neste curso interressa-nos particularizar estes esquemas impondo as seguintes restrições:

(t, n)-partição linear de segredos

1. O segredo $\mathbf{s}$ e todas as quotas z_i são elementos de um corpo $\mathbb{K}$.
2. Cada quota z tem um *titular* identificado por uma marca ("label") $l \in \mathcal{L}$ com $|\mathcal{L}| = n$.
3. Existe uma função PPT implementável $\zeta : \mathcal{L} \longrightarrow \mathbb{K}$ que mapeia marcas em quotas.
4. Existe um algoritmo PPT, M que recebe como input um conjunto de t marcas $\{l_1, \cdots, l_t\}$ e produz uma máscara $\mathbf{b} \in \mathbb{K}^t$ que verifica

$$\mathbf{s} = \sum_{k=1}^{t} b_k \cdot \zeta(l_k) \tag{121}$$

   O algoritmo de recuperação obtém $\mathbf{s}$ usando ζ, M e (121).
5. $\mathbb{K}$, $\mathcal{L}$ e M são informação pública, enquanto que ζ é informação privada.

O exemplo paradigmático é o esquema de Shamir que usa a interpolação polinomial para determinar o segredo $\mathbf{s}$

Partição de segredos (t, n) de Shamir

Seja $\mathbb{K}$ *um corpo e* $f \in \mathbb{K}[x]$ *um polinómio com* $(t-1)$ *raízes distintas tal que* $f(0) = \mathbf{s}$.

$$f(x) \;=\; c_1 + c_2 \cdot x + \cdots + c_t \cdot x^{t-1} \qquad \textit{com} \quad c_1 = \mathbf{s}$$

Seja $\mathcal{L}$ *um conjunto de marcas de cardinalidade* n *e* $H : \mathcal{L} \to \mathbb{K}^*$ *uma função injectiva. As quotas do segredo* $\mathbf{s}$ *são, para cada* $l \in \mathcal{L}$ *, dadas por* $z_l = f(H(l))$.

Dado um conjunto de t marcas $\{l_1, l_2, \ldots, l_t\}$, a recuperação de $\mathbf{s}$ a partir das quotas z, faz-se por simples interpolação polinomial no conjunto de t pontos $\{(x_i, z_i)\}_{i=1}^{t}$ em que $x_i = H(l_i)$ e $z_i = f(x_i)$.

De facto seja $\mathbf{X} \in \mathbb{K}^{t \times t}$ a matriz quadrada definida por $\mathbf{X}_{ij} = x_i^j$. Pela definição tem-se $z_i = \sum_j \mathbf{X}_{ij} \cdot c_j$. Em notação matricial será $\mathbf{X}\,\mathbf{c} = \mathbf{z}$ sendo $\mathbf{c} = (c_1, \ldots, c_t)$ o vector dos coeficientes e $\mathbf{z} = (z_1, \ldots, z_t)$ o vector das quotas.

Portanto tem-se $\mathbf{c} = \mathbf{X}^{-1}\,\mathbf{z}$. Isto significa que

$$\mathbf{s} \;=\; c_1 \;=\; \sum_{i=1}^{t} b_i \cdot z_i$$

em que os coeficientes b_i formam a primeira linha da matriz $\mathbf{X}^{-1}$.

Estas observações sugerem a seguinte implementação do esquema de partilha de segredos (t, n) de Shamir

Protocolo 59 : (t, n) Partilha de segredos de Shamir

Elementos Comuns Um corpo $\mathbb{K}$ e o conjunto de marcas vistas como bit-strings

(1) Um corpo $\mathbb{K}$ e uma função de *hash* $H : \mathbb{B}^* \to \mathbb{K}^*$

(2) Um conjunto finito $\mathcal{L}$ de n marcas $l \in \mathbb{B}^*$

Geração e partilha Um TA gera o segredo $\mathbf{s}$ e gera o segredo e as quotas.

(1) $T : c_i \leftarrow \mathbb{K}^* \quad i = 1 \cdots t \quad ; \quad \mathbf{s} \leftarrow c_1$

(2) $T : x_l \leftarrow H(l) \quad ; \quad z_l \leftarrow \sum_j x_l^j \cdot c_j \quad$ para todo $l \in \mathcal{L}$

Recuperação Dado um conjunto de t marcas $\mathcal{L} = \{l_1, \ldots, l_t\}$ construir a máscara $\mathbf{b}$ que verifica (121). Para isso constrói-se a matriz $\mathbf{X}$ que, depois, é invertida.

(1) $* : x_i \leftarrow H(l_i) \quad i = 1 \ldots t \quad ; \quad \mathbf{X_{ij}} \leftarrow x_i^j \quad i, j = 1 \ldots t$

(2) $* : \mathbf{b} \leftarrow 1^{\text{a}}$ linha de $(\mathbf{X})^{-1}$

Desta forma pode-se estabelecer um protocolo BLS multi-assinatura. São apresentadas duas variantes do passo "assinatura": uma "variante não-anónima" onde é público quem são os assinantes individuais e quais são as assinaturas que eles produzem, e uma variante "assinatura anónima" onde as assinaturas σ_l não são públicas nem é público o conjunto de assinantes.

Assume-se que o esquema de partilha de chaves usa o protocolo 59 com o corpo $\mathbb{K} \equiv \mathbb{Z}_r$. Assume-se que as

quotas são distribuídas por um conjunto de n "provers" identificados por marcas que coincidem com as respectivas identidades. A função de *hash* ID mapeia identidades em elementos de $\mathbb{Z}_r^*$.

Protocolo 60 : BLS multi-assinatura

"Setup" Um TA gera um segredo, distribui as respectivas quotas por um conjunto de "provers" identificados pelas suas identidades e publica a chave pública correspondente.

(1) $T: c_j \leftarrow \mathbb{Z}_r^* \quad , \quad j = 1 \cdots t \quad ; \quad c_1 \rightarrow \beta = g^{c1}$

(2) $T: x_i \leftarrow \mathrm{ID}(\mathcal{P}_i) \quad ; \quad s_i \leftarrow \sum_j x_i^j \cdot c_j$ para todo $i = 1..n$

(3) $\mathcal{P}_i: s_i \leftarrow T.s_i$ para todo $i = 1..n$

Assinatura / variante não-anónima É público o conjunto $\mathcal{A}$ dos t assinantes que tornam públicas as assinaturas individuais. O "verifier" $\mathcal{V}$ constrói a assinatura global.

(1) $\mathcal{P}_l: h \leftarrow H(M) \quad ; \quad s_l \rightarrow \sigma_l = h^{s_l}$ para todo $l \in \mathcal{A}$

(2) $\mathcal{V}$ constrói a máscara $\mathbf{b}$ usando o algoritmo de recuperação no protocolo 59 com o conjunto $\mathcal{A}$

(3) $\mathcal{V}: \sigma \leftarrow \prod_{l \in \mathcal{A}} (\sigma_l)^{b_l}$

Assinatura/ variante anónima Existe um "prover" privilegiado $\mathcal{P}$ que conhece $\mathcal{A}$ e as assinaturas respectivas.

(1) $\mathcal{P}_l: h \leftarrow H(M) \quad ; \quad \sigma_l \leftarrow h^{s_l}$ para todo $l \in \mathcal{A}$

(2) $\mathcal{P}$ constrói a máscara $\mathbf{b}$ usando sobre $\mathcal{A}$ o algoritmo de recuperação no protocolo 59

(3) $\mathcal{P}: \sigma_l \leftarrow \mathcal{P}_l.\sigma_l \quad \forall\, l \in \mathcal{A} \quad ; \quad \cdot \rightarrow \sigma = \prod_{l \in \mathcal{A}} (\sigma_l)^{b_l}$

Verificação Como no protocolo BLS usual.

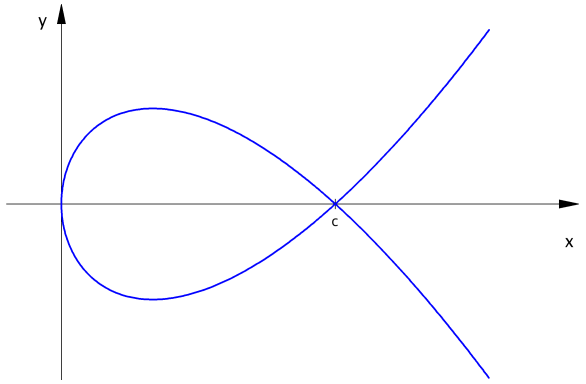
8.Curvas Elípticas

Na procura de grupos cíclicos com as melhores propriedades criptográficas, capazes de aliar garantias de segurança (na perspectiva de dificuldade computacional em resolver os problemas clássicos: DLP, CDHP, BCDHP, etc.) com eficiência de implementação (eficiência na representação e na manipulação computacional), uma área tradicional da Matemática foi "redescoberta": a Geometria Algébrica.

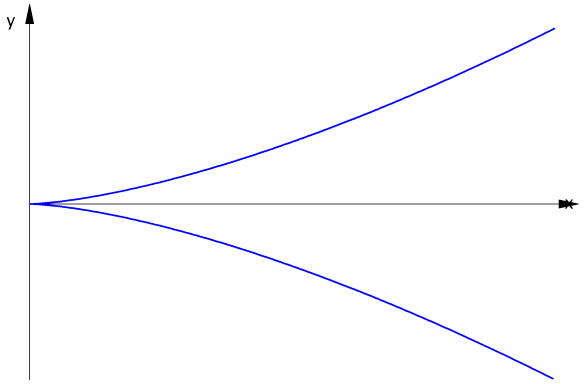
Esta área da Matemática personifica, por um lado, a visão que no século XIX se tinha da Álgebra: o estudos dos polinómios e das sua raízes. Por outro lado dá-lhe a dimensão geométrica e, por isso, estudava essencialmente as curvas definidas em espaços de dimensão real ou racional por equações polinomiais.

Como exemplo considere-se as seguintes curvas definidas no plano $\mathbb{Q}^2$ pelas raízes $\phi(x, y)$ dos polinómios indicados. Note-se que são polinómios a duas variáveis e todos são do 2^o grau em y e do 3^o grau em x.

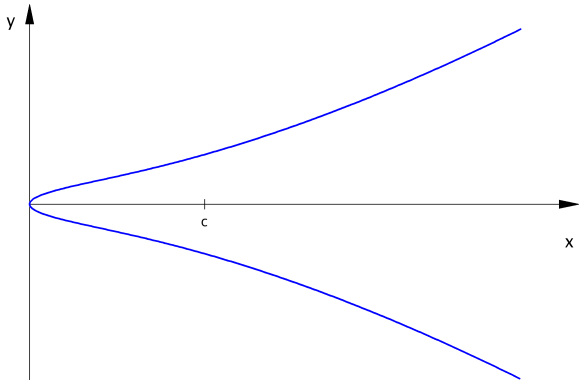
Informalmente, entende-se por *curva* plana racional o conjunto dos pontos $(x, y) \in \mathbb{Q}^2$ para os quais $\phi(x, y) = 0$, i.e, os pontos (x, y) que são *raízes* deste polinómio. Note-se que, ao contrário do que ocorre num polinómio a uma só variável em que as raízes são em número limitado pelo grau do polinómio, para polinómios com mais do que uma variáveis as raízes não estão limitadas pelo grau.



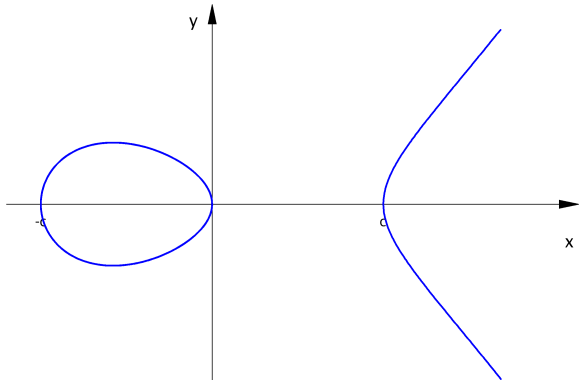
$y^2 - x\,(x-c)^2$



$y^2 - x^3$



$y^2 - x\,(x^2 + c^2)$



$y^2 - x\,(x^2 - c^2)$

Figura 10: Quatro exemplos de curvas planas cúbicas em x e quadráticas em y.

Alguns aspectos importantes que devemos ter em conta:

Estamos habituados a ver este tipo de curvas no plano real $\mathbb{R}^2$; desta forma existem pontos que pertencem à curva no plano real que não estão no plano $\mathbb{Q}^2$. Por exemplo, um ponto de coordenada $x = 1/2$ na curva $y^2 - x(x^2+1)$ tem ordenada $y = \pm\sqrt{5/8}$ que não pertence a $\mathbb{Q}$ (apesar de pertencer a uma extensão algébrica desse corpo). Portanto procurar os pontos em $\mathbb{Q}^2$ da curva $y^2 - x(x^2+1)$, não é uma tarefa trivial.

Na definição de curva, o corpo $\mathbb{Q}$ não tem nada de particular e pode ser substituído por um qualquer outro corpo $\mathbb{K}$. Agora ϕ pertence ao anel dos polinómios a duas variáveis com coeficientes numa extensão de $\mathbb{K}$.

Assim, genericamente, e como primeira definição, pode-se considerar que uma **curva plana** $C/\mathbb{K}$ é o conjunto das raízes em $\mathbb{K}^2$ de um polinómio $\phi \in \mathbb{K}[x, y]$.

Para evidenciar a relação entre a curva, o corpo de suporte e o polinómio, representamos a curva por $C/\mathbb{K}: \phi$.

Note-se que os coeficientes do polinómio estão numa extensão do corpo $\mathbb{K}$ mas não necessariamente em $\mathbb{K}$. Isto faz com que não seja suficiente escolher uma coordenada $x \in \mathbb{K}$ para existir um $y \in \mathbb{K}$ tal que $\phi(x, y) = 0$. De facto pode até acontecer que o polinómio $\phi(x, y)$ não tenha qualquer raiz em $\mathbb{K}^2$

Por exemplo, considere-se um polinómio com coeficientes em $\mathbb{C}$, $\phi(x, y) = i\,y - x - i$ (sendo i a unidade imaginária, $i^2 + 1 = 0$). A curva em $\mathbb{Q}$ definida por este polinómio é formada por um só ponto $\{(0, 1)\}$.

A procura dos pontos de uma curva é, portanto, um processo essencial e, para isso. pode-se recorrer a algumas "heurísticas". Por exemplo, quando o corpo $\mathbb{K}$ é finito, a curva é também um conjunto finito uma vez que o número de raízes de $\phi(x,y)$ em $\mathbb{K}$ está limitado pelo número de pontos disponíveis no plano $\mathbb{K}^2$. De facto, se $\mathbb{K} \equiv \mathbb{F}_q$ for o corpo finito de q elementos, o plano $\mathbb{K}^2$ tem exactamente q^2 possíveis pontos.

Assim é possível, em princípio, encontrar a curva C percorrendo sistematicamente todos $(x,y) \in \mathbb{K}$ e testando, para cada ponto, se verifica $\phi(x,y) = 0$. Obviamente, este procedimento só será computacionalmente viável se q^2 for razoavelmente pequeno.

Algumas formas particulares de polinómio facilitam a construção da curva. Por exemplo, uma classe de curvas importante é a formada pelas *rectas*. No plano $\mathbb{K}^2$ uma **recta** é definida por um polinómio de primeiro grau $l \in \overline{\mathbb{K}}[x,y]$, com $l(x,y) = a\,y + b\,x + c$ sendo $a,b,c \in \overline{\mathbb{K}}$ e $(a \neq 0) \vee (b \neq 0)$.

A curva $C\colon l(x,y)$ goza de uma propriedade muito importante: se tiver dois pontos distintos $P,Q \in \mathbb{K}^2$, com $P = (x_1,y_1)$ e $Q = (x_2,y_2)$, então qualquer $x \in \mathbb{K}$, distinto de x_1 e x_2, ou qualquer $y \in \mathbb{K}$ distinto de y_1 ou y_2, determinam um terceiro ponto $(x,y) \in \mathbb{K}^2$ na mesma curva [52].

[52] Passando a recta pelos pontos $(x,y),(x_1,y_1)$ e (x_2,y_2), tem de se verificar $a\,(y_2-y_1) + b\,(x_2-x_1) = 0$ e $a\,(y-y_1) + b\,(x-x_1) = 0$. Se for $a = 0$ tem-se $x = x_1 = x_2 \in \mathbb{K}$; todo $y \in \mathbb{K}$ determina um ponto em $\mathbb{K}^2$. Se for $a \neq 0$, verifica-se $(y-y_1) = (y_2-y_1)\,(x-x_1)/(x_2-x_1)$; todo $x \in \mathbb{K}$ determina um ponto em $\mathbb{K}^2$.

Quando as rectas se sobrepõem com outras curvas, definidas por polinómios de grau mais elevado, esta propriedade permite dizer
Considere-se, por exemplo, a curva $C\colon y^2 - x^3 - 1$ e a recta $L\colon y - x - 3/4$ representadas na figura 11. Queremos ver que curvas definem em $\mathbb{Q}$; nomeadamente queremos determinar as raízes racionais de $y^2 - x^3 - 1$.

A recta intersecta a curva C em 3 pontos; pela propriedade das rectas, se dois deles tiverem coordenadas racionais o terceiro também tem coordenadas racionais.

Isto sugere um mecanismo para construção de C. Se forem já conhecidos dois pontos de coordenadas racionais em C, traça-se a recta que eles determinam e calcula-se o terceiro ponto de intersecção com a curva. Esse ponto, porque está na recta, também tem coordenadas racionais desde que uma das coordenadas seja racional.

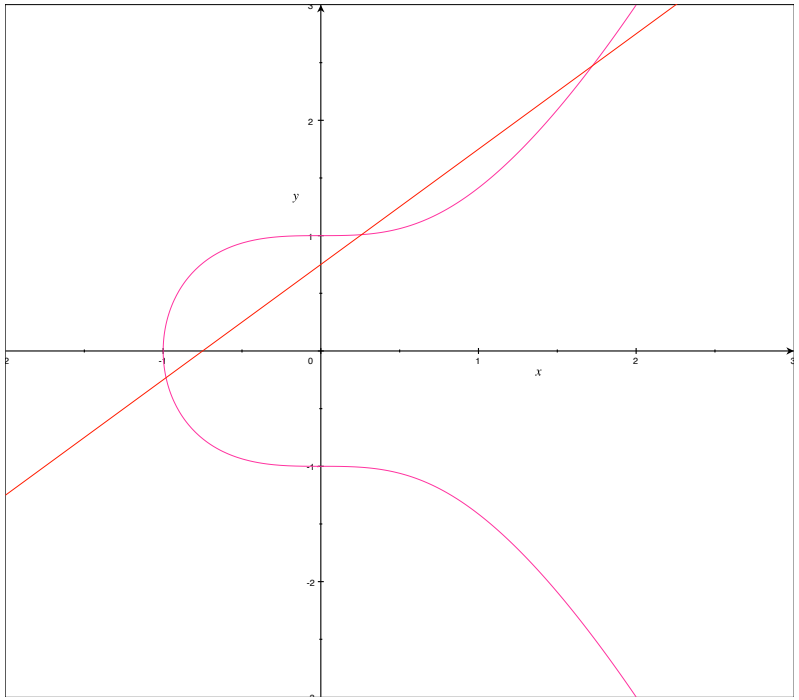


Figura 11: Curvas $y^2 - x^3 - 1$ e $y - x - 3/4$.

Na viabilidade deste mecanismo reside a razão porque se usam este tipo de curvas em Criptografia.

Intersecção de rectas com curvas cúbicas em $\mathbb{Q}$

Considere-se, por exemplo, uma curva cúbica definida pelos pontos $(x, y) \in \mathbb{C}^2$ que verificam a equação

$$y^2 + (a_1\, x + a_3)\, y \;=\; x^3 + a_2\, x^2 + a_4\, x + a_6 \qquad \text{com} \quad a_1, a_2, a_3, a_4, a_6 \in \mathbb{Q} \tag{122}$$

e procuremos determinar os pares (x, y) que pertencem a $\mathbb{Q}^2$.

Para iniciar este procedimento é necessário ter, pelo menos, dois pontos de coordenadas racionais (que podem não ser distintos). Agora a construção de um terceiro ponto a partir de dois outros pontos, $P = (x_1, y_1)$ e $Q = (x_2, y_2)$, passa pela determinação da recta que eles definem e, depois, pelo cálculo da intersecção dessa recta com a curva. Vamos descrever o mecanismo que permite calcular as coordenadas (x_3, y_3), em função das cordenadas de P e Q, do terceiro ponto R de intersecção da recta com a curva.

Se a recta é vertical, que se traduz por ser $x_1 = x_2 \wedge y_1 = -y_2$, o terceiro ponto de intersecção é um ponto especial, designado por *ponto no infinito* e representado por P_∞, que estudaremos na próxima secção.

Quando a recta não é vertical existem parâmetros a determinar $\lambda, \mu \in \mathbb{C}$ tais que todo o ponto (x, y), sobre recta, verifica $y = \mu + \lambda\, x$. Como os três pontos P, Q, R estão sobre a recta, tem-se

$$y_i \;=\; \mu + \lambda\, x_i \qquad \text{para} \quad i = 1, 2, 3 \tag{123}$$

Efectuando a substituição $y \longrightarrow \mu + \lambda\, x$ em (122) obtém-se

$$(\mu + \lambda\, x)^2 + (a_1\, x + a_3)\,(\mu + \lambda\, x) = x^3 + a_2\, x^2 + a_4\, x + a_6$$

Expandindo e agrupando os termos obtém-se

$$x^3 - (\lambda^2 + a_1\, \lambda - a_2)\, x^2 + \ldots \textit{monómios de ordem inferior} \;=\; 0$$

As três soluções desta equação são três ordenadas x_1, x_2, x_3 dos três pontos de intersecção da recta com a curva. Portanto esta mesma equação pode-se também escrever como $(x - x_1)\,(x - x_2)\,(x - x_3) \;=\; 0$. Dado que

$$(x - x_1)\,(x - x_2)\,(x - x_3) \;=\; x^3 - (x_1 + x_2 + x_3)\, x^2 + \ldots \textit{monómios de ordem inferior}$$

conclui-se

$$\lambda^2 + a_1\, \lambda - a_2 \;=\; x_1 + x_2 + x_3 \tag{124}$$

Uma vez que x_1 e x_2 são conhecidos, se λ for conhecido a equação (124) determina x_3. Além disso, sendo λ e x_3 conhecidos, as equações (123) determinam $y_3 = y_1 + \lambda\,(x_3 - x_1)$.

Para determinar λ temos duas situações possíveis:

$P \neq Q$

Sendo $(x_1, y_1) \neq (x_2, y_2)$, e sendo a recta não vertical (o que implica $x_1 \neq x_2$), então as equações (123) conduzem a

$$\lambda = (y_2 - y_1)/(x_2 - x_1) \tag{125}$$

$P = Q$

Neste caso a recta é tangente à curva no ponto (x_1, y_1); portanto λ é o declive da tangente nesse ponto; isto é, $\lambda = [\partial y/\partial x]\,(x_1, y_1)$. Derivando em ordem a x a equação (122), tem-se

$$(2\,y + a_1\,x + a_3)\;(\partial y/\partial x) + a_1\,y = 3\,x^2 + 2\,a_2\,x + a_4$$

Calculando esta derivada no ponto P, conclui-se

$$\lambda \;=\; (3\,x_1^2 + 2\,a_2\,x_1 - a_1\,y_1 + a_4)/(2\,y_1 + a_1\,x_1 + a_3) \tag{126}$$

Através de (125) (quando $P \neq Q$) ou através de (126) (quando $P = Q$) determinamos o parâmetros λ de uma recta não vertical que seja definida pelos dois pontos. Com (124) determinamos

$$x_3 \;=\; \lambda^2 + a_1\,\lambda - a_2 - x_1 - x_2 \qquad , \qquad y_3 \;=\; y_1 + \lambda\,(x_3 - x_1) \tag{127}$$

Estas relações definem o mecanismo computacional que, dados dois pontos racionais P e Q da curva em (122) determina um terceiro ponto racional R que é *colinear* com os dois pontos anteriores.

Note-se que, apesar de um ponto genérico $X = (x, y)$ que verifique a equação (124) ter coordenadas complexas, o mecanismo que acabámos de apresentar assegura que, sendo $P = (x_1, y_1)$ e $Q = (x_2, y_2)$ pontos de coordenadas racionais, o ponto $R = (x_3, y_3)$ também tem coordenadas racionais. De facto os parâmetros λ e μ calculados por (125) ou (126) são racionais e, desta forma, x_3 e y_3, calculados por (127), são necessariamante racionais.

O mecanismo de **colinearidade** determina uma relação ternária entre o três pontos de tal forma que, dados dois deles, é sempre possível calcular o terceiro. Por motivos que serão claros em seguida, vamos escrever essa relação da forma seguinte

$$P \oplus Q \oplus R = P_\infty$$

Para já não vamos dar significado especial ao símbolo "$\oplus$" (que será visto, apenas, como um separador de argumentos) e vamos interpretar "$\cdot = P_\infty$" apenas como um símbolo de predicado ternário. A notação apenas significa que os três pontos são colineares.

Se este fosse o único mecanismo para gerar pontos estaríamos bastante limitados já que, com os três pontos iniciais, o mecanismo permitiria gerar apenas dois pontos adicionais: o ponto $(2, 3)$, que é colinear com os pontos $P = (-1, 0)$ e $S = (0, 1)$, e o ponto $(2, -3)$ colinear com os pontos $(-1, 0)$ e $(0, -1)$.

Por isso são necessários outros mecanismos com esta função. O primeiro deles é óbvio: uma curva que seja definida por um polinómio onde o único termo em y tem grau 2 (um polinómio da forma $y^2 + f(x)$) então se $X = (x, y)$ é raiz do polinómio, também o ponto $(x, -y)$ é raiz do mesmo polinómio. Representamos este ponto por $-X$.

Temos agora uma nova transformação que mapeia pontos racionais da curva noutros pontos racionais da mesma curva: a aplicação $X \mapsto -X$ mapeia o ponto racional (x, y) no ponto racional $(x, -y)$. Esta aplicação designa-se por **simetria**.

O mecanismo da colinearidade parte do princípio que uma recta $y - \mu - \lambda\, x$ contém exactamente 3 pontos da curva $y^2 - x^3 - 1$. A figura 12 ilustra um conjunto de rectas que parecem contrariar esta assumpção.

A recta $y - 2\, x + 1$, que contém R e $-Q$ só parece conter estes dois pontos. A recta horizontal $y = 0$, que contém Q, não contém qualquer outro ponto da curva.

Por outro lado, a recta horizontal $y = 0$ (que passa por P) e as rectas verticais $x = 2$ (que passa por R e $-R$), $x = 0$ (que passa por Q e $-Q$) e $x = -1$ (que passa só por P) parecem conter exclusivamente os pontos indicados.

Tudo depende, porém, da forma como entendemos a noção de "ponto da curva" e como contamos esses pontos.

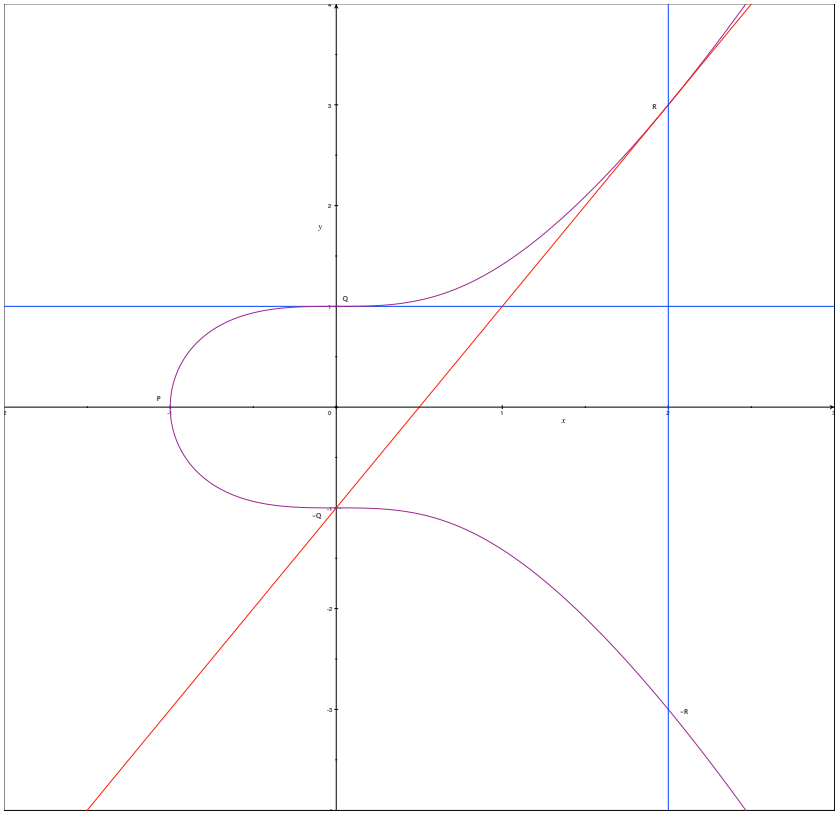


Figura 12: Ponto no infinito na curva $y^2 - x^3 - 1$.

Os pontos comuns a uma recta não-vertical $y + \lambda\, x + \mu = 0$ e à curva $y^2 - 1 - x^3 = 0$ são soluções deste sistema de equações. Substituindo a 1ª equação na segunda obtém-se

$$x^3 - \lambda^2\, x^2 - 2\,\lambda\,\mu\, x - \mu^2 + 1 \;=\; 0 \tag{128}$$

Este polinómio de 3º grau em x tem, no fecho algébrico de $\mathbb{Q}$ (i.e. os complexos $\mathbb{C}$), exactamente 3 raízes distintas. Pode ter uma raiz dupla quando a 1ª derivada também se anula nesse ponto, ou até uma raiz tripla se a 2ª derivada também se anular no mesmo ponto.

Raízes racionais múltiplas do polinómio dão origem a pontos onde a recta é tangente à curva. Quando a raiz é dupla (como no ponto $R = (2, 3)$ para a recta que também passa pelo ponto $-Q = (0, -1)$) interpretamos isso como se a recta intersecta-se duas vezes a curva nesse ponto. A relação de colinearidade deve, neste caso, escrever-se

$$(-Q) \;\oplus\; R \;\oplus\; R \;=\; P_\infty$$

A recta horizontal $y - 1 = 0$ (definida por $\lambda = 0$ e $\mu = -1$) dá origem a um polinómio muito simples; o polinómio (128) reduz-se a x^3 que tem uma raiz tripla no ponto $x = 0$. Neste caso a recta "intersecta" a curva 3 vezes no ponto $Q = (0, 1)$; a relação de colinearidade será, aqui,

$$Q \;\oplus\; Q \;\oplus\; Q \;=\; P_\infty$$

Outra situação deriva da existência de raízes complexas de (128). Por exemplo, a recta horizontal $y = 0$ (definida por $\lambda = \mu = 0$) conduz ao polinómio $x^3 + 1$ que tem uma raiz racional $x_1 = -1$ e duas raízes complexas

$x_2 = -\zeta$ e $x_3 = -\zeta^2$, em que $\zeta \neq 1$ é uma raíz cúbica da unidade[53]. A figura 12 indica apenas o ponto de intersecção $P = (-1, 0)$ definido pela raíz racional; os pontos de intersecção definidos pelas duas raízes complexas, $(-\zeta, 0)$ e $(-\zeta^2, 0)$, não são, aqui, representáveis.

Uma situação distinta ocorre com rectas verticais; tais rectas não podem ser descritas pelo polinómio $y + \lambda\, x + \mu$ [54] mas são descritas, simplesmente, por um polinómio da forma $x - \mu$. Os eventuais pontos racionais comuns à recta e à curva são determinados pelas possíveis raízes quadradas racionais de $1 + \mu^3$ com $\mu \in \mathbb{Q}$. Isto é, serão pontos da forma $(\mu, \pm\sqrt{1 + \mu^3})$ caso μ seja racional e a ráiz quadrada também seja racional.

Portanto uma recta vertical contém, quanto muito, duas raízes racionais do polinómio $y^2 - x^3 - 1$. No entanto, se acrescentar-mos ao conjunto de raízes um ponto extra por onde passam, por definição, todas as rectas verticais, resolve-mos a questão de ter sempre a propriedade da colinearidade estabelecida em triplos de pontos da curva.

Para justificar a introdução do **ponto no infinito** temos de recorrer a algum formalismo de Geometria Algébrica, o que faremos na próxima secção.

Vamos aceitar, para já, que um tal ponto existe, que é representado por P_∞ e por ele passam todas as rectas verticais. Nessa perspectiva a nossa curva vai ser constituída por duas componentes: a primeira é formada pela

[53] As raízes cúbicas complexas da unidade ζ são as raízes em $\mathbb{C}$ do polinómio $X^2 + X + 1$.

[54] Teria que ser $\lambda = \infty$.

raízes racionais do polinómio $\phi(x,y) = y^2 - x^3 - 1$, e se designa-se por "componente afim", e uma segunda componente formada exclusivamente pelo ponto P_∞.

Com esta definição de curva podemos verificar que, pelo menos para este exemplo, duas propriedades importantes:

1. Cada recta intersecta a curva em exatamente 3 pontos, desde que cada ponto conte tantas vezes quantas a respectiva multiplicidade e se entre em conta com o ponto no infinito P_∞ e pontos de coordendas complexas.
2. Cada recta (mesmo que seja vertical), se passa por dois pontos da curva de coordenadas racionais, passa sempre por um terceiro ponto de coordenadas racionais na mesma curva.

Por exemplo, a recta $x = 0$ passa pelos pontos $Q = (0,1)$ e $-Q = (0,-1)$; como é uma recta vertical passa também pelo ponto no infinito P_∞. A colinearidade exprime-se, aqui, por

$$Q \oplus (-Q) \oplus P_\infty = P_\infty$$

A recta vertical $x = -1$ é tangente à curva no ponto $P = (-1,0)$; passa, portanto, duas vezes por esse ponto. Como é vertical passa por P_∞; por isso a colinearidade é

$$P \oplus P \oplus P_\infty = P_\infty$$

8.1 Curvas Planas

A formalização do conceito de curva plana requer algumas noções elementares de Geometria Algébrica. Para não correrm os o risco de enveredar-mos de forma excessiva por uma área da Matemática que, apesar de ser extremamente rica e interessante, tem objectivos que ultrapassam em muito o âmbito deste curso, vamos impor algumas limitações a esse estudo.

Assim, neste curso, vamos entender como "curvas planas" as curvas definidas no espaço bidimensional-dimensional pelas raízes de um polinómio a duas variáveis. Serão apenas estas o objecto do nosso estudo. Procuraremos, desta forma, evitar as complexidades de derivam do estudo das variedades algébricas. Procuraremos também, sempre que possível, usar o chamado sistema de coordenadas afins $\mathbb{A}^2$ e evitar um estudo detalhado de curvas em espaços projectivos.

Essencial ao nosso estudo é não impor limitações ao corpo $\mathbb{K}$ onde vão estar definidas as curvas. Apesar de as intuições geométricas serem mais óbvias em curvas definidas no plano real $\mathbb{R}^2$, não nos podemos esquecer que o nosso objectivo é estudar curvas com interesse criptográfico e isso implica, normalmente, usar outro tipo de corpos, nomeadamente corpos finitos. Como um polinómio de coeficientes no corpo $\mathbb{K}$ tem raízes no seu fecho algébrico $\overline{\mathbb{K}}$, é conveniente pensar, desde o início, em polinómios cujos coeficientes pertencem também a $\overline{\mathbb{K}}$.

Tomemos, então, um corpo $\mathbb{K}$ e $\overline{\mathbb{K}}[x, y]$ o anel dos polinómios a duas variáveis com coeficientes no fecho algébrico $\overline{\mathbb{K}}$ de $\mathbb{K}$. O conjunto dos polinómios $\overline{\mathbb{K}}[x, y]$ tem a estrutura algébrica de um anel. De facto estes polinómios têm

uma estrutura algébrica ainda mais rica: é também um **domínio de factorização única**; isto é, cada elemento do anel pode ser decomposto (de forma única a menos da ordem dos factores) no produto de um número finito de elementos irredutíveis.

□

Curvas planas são conjuntos de pontos que são, de alguma forma, "raízes" de um polinómio irredutível ϕ. Existem dois sistemas possíveis de representar estes pontos: em **coordenadas afins** ou em **coordenadas projectivas**.

Coordenadas Afins

Cada curva é determinada por um polinómio a duas variáveis $\phi(x, y)$ que é irredutível em $\overline{\mathbb{K}}[x, y]$.

Note-se que os coeficientes dos polinómios são elementos do fecho algébrico do corpo $\mathbb{K}$. Note-se também que um polinómio irredutível em $\mathbb{K}[x, y]$ pode não ser irredutível em $\overline{\mathbb{K}}[x, y]$.

Por exemplo, o polinómio $x^2 + 2y^2$ é irredutível em $\mathbb{Q}[x, y]$ mas não é irredutível no anel de polinómios sobre o fecho algébrico. De facto tem-se $(x^2 + 2\,y^2) = (x - i\,\sqrt{2}\,y)\,(x + i\,\sqrt{2}\,y)$ em $\mathbb{C}[x, y]$. Por isso $x^2 + 2\,y^2$ não define uma curva plana no espaço $\mathbb{Q}^2$.

Cada par $(a, b) \in \overline{\mathbb{K}}^2$ determina um ponto P em coordenadas afins. Cada polinómio ϕ mapeia pontos $P \in \overline{\mathbb{K}}^2$ em elementos de $\bar{\mathbb{K}}$ definindo $\phi(P)$ como $\phi(a, b)$. O ponto $P = (a, b) \in \overline{\mathbb{K}}^2$ é **raiz** de ϕ quando $\phi(P) = 0$.

Um polinómio da forma $(x-a)^i\,(y-b)^j$ é um **factor local** em P. O polinómio ϕ é **m-factorizável em P**, se é divisível por um factor local em P de grau m.

135 PROPOSIÇÃO
*Para toda a raíz P de ϕ, existem um inteiro $m \geq 1$ e uma decomposição $\phi = \phi_1 + \cdots + \phi_l$ em que todos os polinómios ϕ_i são m-factorizáveis em P. O maior de tais m designa-se por **multiplicidade** de ϕ em P e representa-se por $\eta_P(\phi)$.*

Este resultado é um corolário de um importante teorema da Álgebra, o Nullstellensatz, que estudaremos com um pouco mais detalhe na secção seguinte.

Note-se que não se exige que todos os polinómios ϕ_i, na decomposição de ϕ, tenham o mesmo factor de grau m. O que tem de ser comum a todas as componentes é o grau do factor e não o próprio factor.

EXEMPLO 32: Considere-se a origem $P = (0,0)$; um factor local em P de grau m é um polinómio da forma $x^i\,y^j$, com $i+j=m$.

Considere-se também o polinómio $\phi = 2\,x\,y + x^3$; obviamente que P é raiz de ϕ. O polinómio é a soma de duas componentes, $2\,x\,y$ e x^3 , ambas 2-factorizáveis em P. A primeira componente tem o factor local $x\,y$; a segunda tem o factor local x^2. Os factores locais em P são distintos, mas ambos têm grau 2.

Qualquer das componentes tem outros factores locais em P: ambas têm factores de grau 1 e a componente x^3 tem um factor de grau 3. Porém o grau 2 é o maior grau que é comum a factores locais em P de ambas as componentes.

Os polinómios x, y e $x+y$ têm todos uma raíz em P de multiplicidade 1. Isto é, $\eta_P(x) = \eta_P(y) = \eta_P(x+y) = 1$. Tem-se $\eta_P(x^2) = \eta_P(y^2) = \eta_P(x\,y) = 2$. Somando um polinómio de multiplicidade 1 com um de multiplicidade 2 (por exemplo, $x + x\,y$) obtém-se um polinómio de multiplicidade 1 em P. O polinómio $2\,x\,y + x^3$ tem, como vimos no exemplo 32, multiplicidade 2 em P.

Como resultado imediato da proposição 135 tem-se

136 TEOREMA
Para toda a raíz P de ϕ, existem polinómios p_{ij}, em que $p_{ij} \neq 0$ implica $p_{ij}(P) \neq 0$, tais que

$$\phi(x,y) \quad = \sum_{i+j=\eta_P(\phi)} (x-a)^i\,(y-b)^j\,p_{ij}(x,y) \tag{129}$$

Se $\eta_P(\phi) > 1$, os polinómios $\partial\phi/\partial x(x,y)$ e $\partial\phi/\partial y$ têm em P uma raíz de multiplicidade $\eta_P(\phi) - 1$. Consequentemente verifica-se $\eta_P(\phi) = 1$ se e só se $\partial\phi/\partial x(P) \neq 0$ e $\partial\phi/\partial y(P) \neq 0$.

A decomposição em (129) pode ser generalizada para polinómios com qualquer número finito de variáveis e, desta forma, pode-se estender a definição de multiplicidade de raíz (proposição 135) para este tipo de polinómios. Por exemplo, se for $\phi \in \mathbb{K}[x,y,z]$ e $P = (a,b,c)$ uma raíz de ϕ em $\mathbb{K}^3$, o polinómio decompõe em $\phi(x,y,z) = \sum_{i+j+k=m} (x-a)^i\,(y-b)^j\,(z-c)^k\,p_{ijk}(x,y,z)$; a multiplicidade de ϕ em P é o maior m para o qual existe esta decomposição de ϕ.

□

Nem todos os pontos das curvas são definidos por pares $(a, b) \in \overline{\mathbb{K}}^2$. Nomeadamente o comportamento assimptotico de curvas é expresso pela existência dos chamados "pontos no infinito".

Considere-se o caso simples das rectas; sabemos que uma recta no plano pode ser determinada por dois pontos distintos ou, em alternativa, por um ponto e um declive ("direcção"). Numa recta o declive pode ser infinito (se a recta for vertical) ou então, sendo finito, é um elemento de $\overline{\mathbb{K}}$.

O polinómio para uma recta que passe pelos pontos (a, b) e (a', b') é $(x - a)(b - b') - (y - b)(a - a')$. O polinómio para a recta que passa pelo ponto (a, b) tem declive μ, exige um pouco mais cuidado: se o declive for infinito (recta vertical) o polinómio é $(x - a)$; se μ for finito, o polinómio é $\mu(x - a) - (y - b)$.

Idealmente deveríamos ter apenas uma situação: uma recta é definida por dois pontos. Para isso, e para tentar unificar estas três situações, os matemáticos do século XVII introduziram a noção de **pontos no infinito**.

Nesta perspectiva cada declive μ (finito ou infinito) introduz um ponto no infinito P_μ; diz-se que uma recta passa pelo ponto P_μ se e só se tem declive μ. Uma curva C passa pelo ponto P_μ se tem uma assimptota com declive μ.

A unificação completa destas representações e um sistema de pontos que contenha os pontos no infinito só pode ser feito recorrendo às coordenadas projectivas. No entanto, mesmo nas coordenadas afins, interessa-nos ver o papel dos pontos no infinito na caractarização do comportamento assimptótico de curvas.

137 NOÇÃO

A **homogenização** *de* $\phi \in \mathbb{K}[x,y]$ *com grau total* d*, é o polinómio* $\phi_h \in \mathbb{K}[x,y,z]$ *tal que*

$$\phi_h(x,y,z)/z^d \;=\; \phi(x/z, y/z) \tag{130}$$

Diz-se que $\phi(x,y)$ *tem uma raiz de multiplicidade* m *em* P_∞ *quando* $\phi_h(z,y,z)$*, tem uma raiz de multiplicidade* m *em* $(0,1,0)$*; identicamente, para* μ *finito, diz-se que* ϕ *tem uma raiz de multiplicidade* m *em* P_μ *quando* $(1,\mu,0)$ *for uma raiz de multiplicidade* m *de* $\phi_h(z,y,z)$*.*

Tento em atenção que $\phi(x,y) = \phi_h(x,y,1)$, constata-se que as raizes (x,y) de ϕ são precisamente as raízes de ϕ_h da forma $(x,y,1)$. Portanto ϕ_h captura não só todas as raízes afins de ϕ como também as raízes no infinito. Este incremento em representatividade tem, obviamente, um custo: a variável adicional z. Para ϕ, as raízes procuram-se num espaço a duas dimensões; ao invés as raízes de ϕ_h procuram-se num espaço a três dimensões.

EXEMPLO 33:

1. Uma recta $\phi = a\,x + b\,y + c$ homogeniza em $\phi_h = z\,(a\,x/z + b\,y/z + c) = a\,x + b\,y + c\,z$. Temos $\phi_h(0,1,0) = b$ e $\phi_h(1,\mu,0) = a + b\,\mu$. Portanto P_∞ é raíz da recta se e só se $b = 0$; i.e., se a recta é vertical e passa pelo ponto $x = -c/a$. Se $b \neq 0$, P_μ é raíz da recta se for $\mu = -a/b$.
2. O polinómio $\phi = y^2 + x^3 + x\,y + 1$ tem grau total é 3 e a sua homogenização é

$$\phi_h \;=\; z^3\,((y/z)^2 + (x/z)^3 + (x/z)\,(y/z) + 1) \;=\; y^2\,z + x^3 + x\,y\,z + z^3$$

Tem-se $\phi_h(0,1,0) = 0$ e $\phi_h(1,\mu,0) = 1$; portanto ϕ tem P_∞ como raíz de ϕ mas nenhum P_μ, com μ finito, é raíz.

Não é possível construir ϕ_h como uma soma de múltiplos de monómios $x^i\,(y-1)^j\,z^k$ cujo grau total $i+j+k$ seja 2 ou superior; assim a multiplicidade da raíz P_∞ é, apenas, 1.

3. Considere-se finalmente $\phi = x^2\,y + x$ cujo grau total é 3 e tem homogenização $\phi_h = x^2\,y + x\,z^2$. Tem-se $\phi_h(0,1,0) = 0$ e $\phi_h(1,\mu,0) = \mu$. Portanto P_∞ e P_0 são ambas raízes no infinito de ϕ.

   Claramente, a multiplicidade de ϕ_h é 2 em $(0,1,0)$ (atente-se à forma $\phi_h = x^2\,p + z^2\,q$, com $p = y$ e $q = x$) e é 1 em $(1,0,0)$ (atente-se à forma $\phi_h = (x-1)\,p + y + z\,q$, com $p = (x+1)\,y$ e $q = x\,z$).

138 NOÇÃO

A **curva plana** *em* $\mathbb{A}^2$, *definida por um polinómio* $\phi \in \mathbb{K}[x,y]$ *que é irredutível em* $\overline{\mathbb{K}}\,[x,y]$, *é o conjunto formado pelas raízes afins ou no infinito de* ϕ. *Um* **ponto singular** *é uma raíz de* ϕ *com multiplicidade* > 1. *A curva diz-se* **não-singular** *se não contém pontos singulares. Se* K *é uma qualquer extensão de* $\mathbb{K}$, *os pontos* K**-racionais** *da curva são os pontos afins de coordenadas* $(x,y) \in K^2$.

Notas

1. **Curvas Triviais**

   Os polinómios 1 e 0 são ambos irredutíveis e definem duas "curvas" triviais. O polinómio 1 não tem qualquer raíz; por isso, a "curva" é o conjunto vazio de pontos $\emptyset$. O polinómio 0, ao invés, tem como raizes qualquer ponto $(x,y) \in \overline{\mathbb{K}}^2$ e qualquer ponto no infinito; é o espaço total que representamos por $\mathbb{P}^2$.

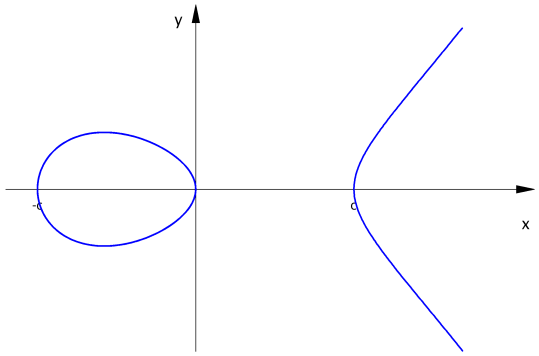
2. **Pontos Singulares**

   Para detectar pontos singulares pode-se usar o teorema 136 e o critério das derivadas parciais.

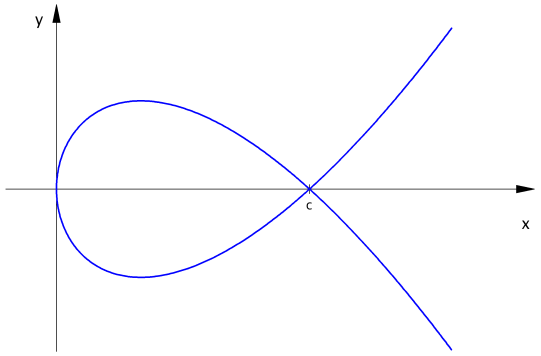
   Por exemplo, cúbica $\phi = y^2 - x\,(x^2-1)$ define uma curva plana formada por todos os pontos (x,y) que são raízes deste polinómio e ainda pelo ponto P_∞ já que, se verifica facilmente, o polinómio tem essa raiz no infinito.

Note-se que $\partial\phi/\partial x = 1 - 3\,x^2$ e $\partial\phi/\partial y = 2\,y$; os únicos pontos que são raízes de ambas as derivadas são $(\pm\sqrt{1/3}, 0)$. Porém nenhum destes pontos pertence à curva; por isso ela é não-singular.

Já o polinómio $\phi' = y^2 - x\,(x-1)^2$ tem derivadas parciais $\partial\phi'/\partial y = 2\,y$ e $\partial\phi'/\partial x = (x-1)\,(1-3\,x)$. As raízes comuns a ambas estes dois polinómios são os pontos $(1,0)$ e $(1/3,0)$. Note-se que $(1,0)$ é um ponto da curva; por isso ela é singular.



$y^2 - x\,(x^2-1)$



$y^2 - x\,(x-1)^2$

3. **Pontos Racionais**

É preciso ter em conta que as raízes afins de $\phi \in \mathbb{K}[x,y]$ podem ter coordenadas fora do corpo $\mathbb{K}$. Tome-se, por exemplo, $\mathbb{K} \equiv \mathbb{Q}$ e considere-se $\phi = y^2 - x^3 - 1$. Genericamente as raízes afins de ϕ têm coordendas complexas, uma vez que $\overline{\mathbb{Q}} \equiv \mathbb{C}$.

Fixe-se um racional qualquer b e procure-se pontos afins da curva da forma (a,b). O valor de a tem de ser raíz do polinómio $x^3 - (b^2-1)$. Se for $b^2 \neq 1$ existem três raízes distintas deste polinómio: um valor algébrico, $a = \sqrt[3]{b^2-1}$, e dois valores complexos $a\,\zeta$ e $a\,\zeta^2$, sendo ζ uma raíz cúbica complexa da unidade. Se for $b^2 = 1$ o polinómio tem uma raíz tripla em 0. A menos deste último caso, as raízes de ϕ da forma (a,b), com b racional, muito provavelmente não têm uma cordenada a que seja racional: duas são complexas e uma é algébrica, provavelmente irracional.

Existem, no entanto, raízes racionais do polinómio $x^3 - (b^2-1)$, para determinados valores de b. Por exemplo, para $b = 0$, temos

uma raíz racional $a = -1$; para $b = 3$ temos a raíz $a = 2$, etc. Estes pontos, $(-1, 0), (2, 3)$, etc, são pontos racionais da curva.

Seja C a curva plana determinada pelo polinómio ϕ; esse facto denota-se por $C : \phi$. Uma maneira de interpretar a curva C é através do conjunto formado por todos os polinómios que se anulam em todos $P \in C$.

$$\mathbf{I}(C) \;=\; \{\, f \in \overline{\mathbb{K}}\,[x, y] \;\mid\; f(P) = 0 \quad \text{para todo} \quad P \in C \,\} \tag{131}$$

É fácil verificar que o conjunto $\mathbf{I}(C)$ é um ideal; isto é, é fechado por somas e por multiplicação por um qualquer polinómio. O facto de ϕ ser irredutível em $\overline{\mathbb{K}}\,[x, y]$ assegura que o ideal é primo; isto é, se $f \cdot g$ pertence ao ideal, um dos polinómios f ou g tem de pertencer ao ideal. Veremos adiante (ver noção 157, página 474) uma exposição sucinta da noção de ideal e suas aplicações à Teoria das Curvas.

O anel quociente $\overline{\mathbb{K}}\,[x, y]/\mathbf{I}(C)$ identifica como equivalentes dois polinómios que são iguais em todos os pontos da curva; isto é, $p \sim q$ sse $p - q \in \mathbf{I}(C)$ ou, equivalentemente, sse $p(P) = q(P)$ para todo $P \in C$. Este anel representa-se por $\mathbb{A}(C)$ e designa-se por **anel afim** ou **anel de coordenadas** da curva C.

As noções de m-factorização e multiplicidade podem ser estendidas a

139 NOÇÃO
Seja ϕ um polinómio que tem uma raíz P sobre uma curva C. Representamos por $\eta_P(\phi; C)$, e designa-se por

multiplicidade *de ϕ em P sobre C, o maior m para o qual existe um polinómio $u \in \mathbf{I}(C)$ tal que $\phi - u$ tem uma raíz de multiplicidade m em P.*

Quando $\eta_P(\phi; C) = 1$, então ϕ ***intersecta*** *a curva C em P; se $\eta_P(\phi; C) > 1$, ϕ é* ***tangente*** *a C em P.*

Comparando com a noção de multiplicidade simples (proposição 135) vemos que a mudança essencial está no facto de não se exigir que ϕ tenha multiplicidade m em P mas, em vez disso, exigir-se que a diferença $\phi - u$, para algum $u \in \mathbf{I}(C)$, tenha essa multiplicidade. Desta forma a multiplicidade de ϕ em P, $\eta_P(\phi)$, é equivalente à multiplicidade $\eta_P(\phi; 0)$ de ϕ em P sobre a curva trivial definida pelo polinómio 0.

EXEMPLO 34: Considere-se a recta $\phi = (y - 1)$. Seja C a curva definida pelo polinómio $\psi = y^2 - 1 - x^3$. O ponto $P = (0, 1)$ é raíz de ϕ e de ψ; por isso é um ponto de C comum com a curva definida por ϕ.

Com um pouco de manipulação pode-se constatar que

$$(y - 1) - \frac{1}{4}(3 - y)(y^2 - 1 - x^3) = \frac{1}{4}(3 - y)x^3 + \frac{1}{4}(y - 1)^3$$

O lado direito da igualdade é um polinómio com uma raíz em P de multiplicidade 3 (atente-se aos factores locais x^3 e $(y - 1)^3$). O lado esquerdo é uma diferença da forma $\phi - u$ para um polinómio $u = \frac{1}{4}(3 - y)\psi$ que, por ser múltiplo de ψ, é um elemento de $\mathbf{I}(C)$.

Consequentemente, atendendo à definição, o polinómio $y - 1$ tem uma multiplicidade 3 em P sobre a curva C. De facto $(y - 1)$ representa uma recta tangente à curva C onde o ponto de contacto P tem multiplicidade 3.

A noção de intersecção ou contacto de duas curvas é caracatrizada por um importante teorema[55].

140 TEOREMA (BEZOUT)
Sejam $C:\phi$ e $D:\psi$ duas curvas distintas. Então $C \cap D$ é um conjunto finito e verifica-se

$$\sum_{P \in C \cap D} \eta_P(\phi;\psi) \quad = \quad \sum_{P \in C \cap D} \eta_P(\psi;\phi) \quad = \quad \deg(\phi) * \deg(\psi) \tag{132}$$

É importante ter-se em atenção que nas curvas C e D estão incluídos não só os pontos afins como os pontos no infinito. Se uma das curvas (por exemplo $C:\phi$) for uma recta, tem grau 1 e, por isso, a soma (132) é igual ao grau do polinómio ψ. Isso significa que uma recta contacta uma curva ψ em tantos pontos (incluindo os pontos no infinito e contando cada ponto tantas vezes quantas a sua multiplicidade) quantos o grau de ψ.

EXEMPLO 35:
Considere-se a curva elíptica $\psi = y^2 - x^3 - 1$. Como o grau de ψ é 3, o teorema de Bézout diz-nos que qualquer recta ϕ contacta a curva em exactamente 3 pontos.

Por exemplo, recta $\phi = y$ intersecta a curva em 3 pontos distintos: o ponto racional $(-1,0)$ e dois pontos de ordenada complexa $(-\zeta,0)$ e $(-\zeta^2,0)$, sendo ζ uma raíz cúbica, complexa da unidade. Todos eles têm multiplicidade 1.

[55] Para prova ver HARTSHORNE, *Algebraic Geometry*.

A recta $\phi = y - 1$ contacta a curva ψ no ponto $P = (0, 1)$ e, como vimos no exemplo 34, a multiplicidade do contacto é 3. Portanto esta recta não contacta a curva em qualquer outro ponto.

A recta $\phi = x$ contacta a curva em dois pontos afins $(0, 1)$ e $(0, -1)$ e ainda no ponto do infinito P_∞.

Coordenadas Projectivas

Desde pelo menos o século XVII que os matemáticos se aperceberam que, adicionando certos pontos fictícios a rectas e outras curvas, a geometria Euclidiana poderia ser muito simplificada. Como exemplo, considere-se um par de asserções duais da geometria plana clássica:

(1) Duas rectas distintas determinam um único ponto: o seu ponto de intersecção.

(2) Dois pontos distintos determinam uma única recta: a recta que passa por ambos os pontos.

A asserção (1) não é válida quando as rectas são paralelas; esta excepção pode ser resolvida assumindo que rectas contêm um "ponto no infinito" e que as rectas paralelas se intersectam nesse "ponto no infinito"; a asserção, agora, é universalmente válida.

Para que a 2ª asserção continue válida com a introdução dos pontos no infinito temos de assumir que um ponto no plano e um ponto no infinito determinam também uma única recta. Isto faz supor que "ponto no infinito" seja equivalente ao conceito de "direcção" ou "inclinação" das rectas: um ponto no plano e uma directação determinam, realmente, uma única recta. Do mesmo modo, para que (2) continue a ser válida com dois pontos no infinito

distintos (duas direcções diferentes), somos levados naturalmente à conclusão que todos os pontos no infinito estão colocados sobre uma mesma recta e que tal recta só contém pontos no infinito; isto é, existe uma recta totalmente situada no infinito.

Estes conceitos têm resultados algébricos importantes; no entanto, em coordenadas afins, são difíceis de visualizar e conduzem a noções pouco naturais; por exemplo, pontos que são direcções. As coordenadas projectivas apareceram nos princípios do século XIX para ser possível lidar facilmente com este tipo de situações sem ter necessidade de introduzir interpretações "estranhas" para certos pontos, rectas ou outras curvas e todas estas entidades serem representados de uma única forma.

Esta representação unificada exige uma representação das entidades (pontos, rectas e curvas) segundo vários pontos de vista que são, de alguma forma, equivalentes. Nomeadamete, para representação de pontos, não basta apenas um tuplo de coordenadas (como nas coordenadas afins) mas vários tuplos ligados por uma relação de equivalência.

141 Noção

Representa-se por $\mathbb{P}^2$ *o conjunto das rectas em* $\mathbb{K}^3$ *que passam pela origem. Os elementos de* $\mathbb{P}^2$ *designam-se por* ***pontos projectivos*** *ou* ***pontos em coordenadas projectivas*** *de dimensão 2.*

Cada recta em $\mathbb{K}^3$ que passa pela origem é determinada por por um polinómio da forma $a\,x + b\,y + c\,z$ em que pelo menos um dos coeficientes (a, b, c) é diferente de zero. Note-se que a mesma recta pode ser representada por outro polinómio $a'\,x + b'\,y + c'\,z$ desde que se verifique $a = \lambda\,a' \wedge b = \lambda\,b' \wedge c = \lambda\,c'$ para algum $\lambda \neq 0$.

Esta observação conduz-nos a uma forma alternativa de definir $\mathbb{P}^2$ através de uma relação de equivalência sobre triplos de coordenadas. Considere-se triplos $P = (a, b, c)$ e $Q = (a', b', c')$ em $\mathbb{K}^3 \setminus \{(0, 0, 0)\}$ (i.e., pelo menos uma das componentes de cada tuplo é $\neq 0$); defina-se a a seguinte relação nesse espaço

$$P \sim Q \quad \Leftrightarrow \quad (\exists \lambda \neq 0)\, [a = \lambda a' \ \wedge \ b = \lambda b' \ \wedge \ c = \lambda c'] \tag{133}$$

A relação $\sim$ é claramente uma relação de equivalência. Os pontos em coordenadas projectivas são as classes de equivalência definidas em $\mathbb{K}^3 \setminus \{(0, 0, 0)\}$ por esta relação de equivalência.

142 NOÇÃO

No contexto de $\mathbb{P}^2$ *os* **pontos afins** *são as classes de equivalência que contêm triplos da forma* $(x, y, 1)$. *Os* **pontos no infinito** *são classes de equivalência que contém triplos da forma* $(x, y, 0)$, *em que* $x \neq 0$ *ou* $y \neq 0$; *nomeadamente* P_∞ *designa o ponto determinado pelo triplo* $(0, 1, 0)$ *e, para cada* $\mu \in \overline{\mathbb{K}}$, P_μ *designa o ponto no infinito determinado pelo triplo* $(1, \mu, 0)$.

Um polinómio $\phi \in \overline{\mathbb{K}}\,[x, y, z]$ em que todos os monómios têm o mesmo grau d diz-se **homogéneo** de grau d. Um tal polinómio verifica $\phi(\lambda\, x, \lambda\, y, \lambda\, z) = \lambda^d\, \phi(x, y, z)$ para todo λ e todo triplo (x, y, z). Por isso se um triplo (x, y, z) é raiz do polinómio, qualquer outro triplo que lhe seja equivalente é também raiz do polinómio.

Um ponto P é **raiz** de um polinómio homogéneo quando existe um representante desta classe que é raiz do polinómio. Se tal acontecer, então (como vimos) qualquer outro triplo que lhe seja equivalente é também raiz do mesmo polinómio. Nestas circunstância escreve-se $\phi(P) = 0$.

Por exemplo, P_∞ é raíz do polinómio homogéneo $z\,y^2 + x^3 + x\,z^2$.

143 NOÇÃO

Um polinómio homogéneo $\phi \in \mathbb{K}[x, y, z]$ *que seja irredutível no fecho algébrico* $\overline{\mathbb{K}}\,[x, y, z]$ *determina uma* **curva plana** *em coordenadas projectivas (ou* **curva projectiva***) definida como o conjunto das raízes desse polinómio. A curva é* **singular** *quando existe um ponto da curva que é raiz, simultaneamente, das três derivadas parciais* $\partial\phi/\partial x$, $\partial\phi/\partial y$ *e* $\partial\phi/\partial z$.

Na representação de pontos, a vantagem das coordenadas projectivas está no facto de situações exceptionais (como o ponto no infinito) não exigirem nenhum tratamento especial; todos os pontos são referenciados do mesmo mod. A desvantagem está no facto de precisarmos de 3 coordenadas (em vez de 2) para definir o ponto e esse triplo de coordenadas ser apenas um representante da classe de equivalência que determina o ponto. Isto significa que qualquer propriedade que quisermos mostrar para um ponto tem de ser invariante pela multiplicação das coordenadas por um factor de escala $\lambda \neq 0$ arbitrário.

Uma consequência desta exigência é o facto de apenas se poder usar polinómios homogéneos. Enquanto que nas coordenadas afins qualquer polinómio irredutível definia uma curva, nas coordenadas afins só os polinómios homogéneos definem curvas.

□

Vamos colocar de novo a questão de curvas projectivas em $\mathbb{P}^2$ sobre um corpo algebricamente fechado K.

Como vimos na noção 143 na página 463, cada curva é determinada por um polinómio homogéneo e irredutível $\phi \in K[x,y,z]$ irredutível[56]. Dada uma curva $C:\phi$, representa-se por $\mathbf{I}(C)$ o seu ideal

$$\mathbf{I}(C) \;=\; \{\, p \in K\,[x,y,z] \;\mid\; p(P)=0 \text{ para todo } P \in C \,\} \tag{134}$$

Como consequência do Nullstellensatz, do facto de K ser algebricamente fechado e ϕ ser irredutível, tem-se

144 FACTO
Tem-se $p \in \mathbf{I}(C)$ se e só se p é divisível por ϕ.

Quando passamos à segunda parte desta definição (a noção de curva não-singular) surge a exigência de as três derivadas principais de ϕ não se anularem simultânemamente em nenhum ponto da curva. Para vermos o alcance desta restricção, é importante ver o seguinte morfismo e os resultados seguintes.

145 NOÇÃO
Seja C uma curva projectiva não-singular determinada por um polinómio homogéneo ϕ; seja d o seu grau. O morfismo $\mathcal{J}: C \longrightarrow \mathbb{P}^2$ determinado pelo triplo de polinómios homogéneos de grau $d-1$

$$\mathcal{J} \;=\; [\,\partial\phi/\partial x \;,\; \partial\phi/\partial y \;,\; \partial\phi/\partial z\,] \tag{135}$$

designa-se por ***jacobiano*** *de C.*

[56] Atente-se que, neste caso, K coincide com o seu fecho algébrico.

Porque C é não-singular, em qualquer zero do polinómio ϕ pelo menos uma das três componentes de $\mathcal{J}^C$ não se anula. Por isso $\mathcal{J}$ define realmente um morfismo.

146 TEOREMA
Seja $\mathcal{J}$ o jacobiano da curva projectivas C; então, imagem $\mathcal{J}^C(C)$ define em $\mathbb{P}^2$ uma curva projectiva que designamos por **curva dual** *de C.*

147 LEMA *Se $\phi \in K[x,y,z]$ é um qualquer polinómio homogéneo de grau d, verifica-se*

$$x\,\partial\phi/\partial x + y\,\partial\phi/\partial y + z\,\partial\phi/\partial z \quad = \quad d\cdot\phi \tag{136}$$

Prova O polinómio pode-se escrever como $\phi = \sum_{i+j+k=d} a_{ijk}\,x^i\,y^j\,z^k$. Temos

$$x\,\partial\phi/\partial x \;=\; \sum_{i+j+k=d} i\cdot a_{ijk}\,x^i\,y^j\,z^k$$

e formas análogas para $y\cdot\partial\phi/\partial y$ e $z\cdot\partial\phi/\partial z$. Donde

$$x\,\partial\phi/\partial x + y\,\partial\phi/\partial y + z\,\partial\phi/\partial z \;=\; \sum_{i+j+k=d} (i+j+k)\cdot\,a_{ijk}\,x^i\,y^j\,z^k \;=\; d\cdot\phi$$

Curvas definidas por polinómios do 1º grau, $a\,x + b\,y + c\,z$ designam-se por **rectas projectivas**.

Claramente, cada triplo $(a, b, c) \in \mathbb{K}^3$ determina uma recta projectiva a menos da relação de equivalência nos pontos de $\mathbb{P}^2$; isto é, os triplos (a, b, c) e $(\lambda\, a, \lambda\, b, \lambda\, c)$, com $\lambda \neq 0$, determinam exactamente a mesma recta. Consequentemente

148 FACTO
Existe um isomorfismo entre $\mathbb{P}^2$ *e o conjunto de todas as rectas projectivas em* $\mathbb{P}^2$*, isomorfismo esse que ao ponto* $P = [a, b, c]$ *faz corresponder a recta definida por* $a\, x + b\, y + c\, z$.

A recta projectiva (e o respectivo polinómio homogéneo de 1º grau) determinados por $P \in \mathbb{P}^2$ são, aqui, representados por $\mathbf{l}(P)$.

149 NOÇÃO
A recta $\mathbf{l}(\mathcal{J}^C(P))$ *designa-se por* **tangente** *à curva* C *no ponto* P.

□

No espaço afim $\mathbb{A}^3$ uma curva projectiva C determina uma superfície cónica com vértice na origem. Considere-se o ideal $\mathbf{I}(C)$ e o anel afim $\mathbb{A}^3(C)$. Recorde-se que este anel é definido como o quociente $K\,[x, y, z]\,/\mathbf{I}(C)$.

No anel afim $\mathbb{A}^3(C)$, a noção de multiplicidade de um polinómio p num ponto $P \in \mathbb{A}^3$ sobre a superfície C é definido da forma usual.

Sumariamente: um **factor local** de $P = (a, b, c)$ é um polinómio da forma $(x-a)^i\,(y-b)^j\,(z-c)^k$; p é m-**factorizável** em P se é divisível por um factor local em P de grau m; a **multiplicidade** de p em P é o maior $m \geq 0$ tal que p é decomponível numa soma de polinómios m-factorizáveis em P. O Nullstellensatz assegura que p tem uma raíz em P se e só se tem multiplicadade maior que zero nesse ponto.

Finalmente, se $P \in C$, a **multiplicidade** de $p \notin \mathbf{I}(C)$ em P **sobre** C, representada por $\eta_P(p; C)$, é a maior multiplicidade em P de polinómios u tais que $p - u \in \mathbf{I}(C)$.

Se $p(P) \neq 0$, convencionamos que $\eta_P(p; C) = 0$. Se $p \in \mathbf{I}(C)$ convencionamos que $\eta_P(p; C) = \infty$.

150 Proposição
Seja $p \in K[x, y, z]$ um polinómio homogéneo e $C : \phi$ uma curva projectiva. Então, para todos $P = (a, b, c) \in K^3$ e $\lambda \neq 0$, tem-se $\eta_P(p; C) = \eta_{\lambda P}(p; C)$.

Prova Se $\eta_P(p; C) = m$ então existem polinómios u, v tais que $p = u\,\phi + v\,f$ sendo f um factor local em $P = (a, b, c)$ de grau m. Isto é, $f = (x-a)^i\,(y-b)^j\,(z-c)^k$ com $i + j + k = m$. Seja H a aplicação que mapeia qualquer polinómio $h(x, y, z)$ em $\lambda^m\,h(x/\lambda, y/\lambda, z/\lambda)$. Se h for homogéneo de grau d tem-se $H(h) = \lambda^{m-d}\,h$. Verifica-se facilmente que $f' = H(f)$ é um factor local em λP de grau m.

Consequentemente $H(p) = H(u)H(\psi) + H(v)\,H(f)$ conduz à igualdade $p = u'\,\phi + v'\,f'$ já que tanto p como ϕ são homogéneos. Consequentemente a multiplicidade de p sobre C em λP é, pelo menos, m. Dada a simetria da afirmação, terá de ser exactamente m.

Uma multiplicidade importante é a multiplicidade da tangente a uma curva no ponto de tangência e sobre a curva. Em consequência do lema 147, a tangente à curva C no ponto P, contém sempre esse ponto P. Portanto a tangente intersecta a curva. De facto, tem-se

151 LEMA *Seja $t_P = \mathbf{l}(\mathcal{J}^C(P))$ a recta tangente à curva C no ponto P. Então tem-se sempre $\eta_P(t_P; C) > 1$.*

EXEMPLO 36: Considere-se a curva C definida por $y^2\,z - x^3 - z^3$. São pontos da curva,

$$P = [0,1,1] \qquad Q = [1,0,1] \qquad P_\infty = [0,1,0]$$

Pretende-se determinar as tangentes nesses pontos assim como a multiplicidade respectiva. Nesta curva tem-se

$$\mathfrak{D} \;=\; [-3\,x^2, 2\,y\,z, y^2 - 3\,z^2] \qquad \mathcal{J}^C(P) = [0,-1,1] \qquad \mathcal{J}^C(Q) = [1,0,1] \qquad \mathcal{J}^C(P_\infty) = [0,0,1]$$

As tangentes respectivas são as rectas

$$t_P \;=\; z - y \qquad t_Q \;=\; x + z \qquad t_{P_\infty} \;=\; z$$

Como o polinómio tem grau 3 e as rectas tangente têm grau 1, o teorema de Bézout (teorema **??**) diz-nos que a multiplicidade não excede 3. O lema 151 diz-nos que a multiplicidade é > 1.

As funções racionais em coordenadas projectivas são definidas, também, através de fracções de polinómios homogéneos com o mesmo grau.

152 NOÇÃO
Seja C/K uma curva projectiva em $\mathbb{P}^2$ e K um corpo algebricamente fechado. Um par de polinómios $f, g \in K[x, , y, z]$ determina uma **fracção homogénea** *quando ambos são homogéneos e têm o mesmo grau.*

O espaço $K(C)$ *das* ***funções racionais*** *sobre* C *é o espaço quociente definido no conjunto das fracções homogéneas pela relação de equivalência*

$$f/g \sim p/q \quad \Leftrightarrow \quad f\,q - g\,p \in \mathbf{I}(C)$$

153 DEFINIÇÃO
A ***ordem*** *de* $f = p/q \in \mathbb{K}(C)^*$ *(isto é,* $f \neq 0$*) em* P*, representado por* $\mathsf{ord}_P(f)$ *, é a diferença*

$$\mathsf{ord}_P(f) \;=\; \eta_P(p;C) - \eta_P(q;C)$$

Se for $\mathsf{ord}_P(f) > 0$ *diz-se* f *tem um* ***zero*** *de ordem* $\mathsf{ord}_P(f)$ *em* P*; se for* $\mathsf{ord}_P(f) < 0$ *diz-se que* f *tem um* ***pólo*** *de ordem* $-\mathsf{ord}_P(f)$ *em* P*.*

Por convenção, a função racional nula $f = 0$ *tem um zero de ordem* ∞ *em todo o ponto de* C*.*

É fácil verificar que estas noções são independentes do representante p/q escolhido para a função racional f. Tem-se também, para todo o par de funções racionais $f, g \in \bar{\mathbb{K}}(C)^*$ e todo o ponto P da curva C

$$\mathsf{ord}_P(f\,g) \;=\; \mathsf{ord}_P(f) + \mathsf{ord}_P(g) \tag{137}$$

O seguinte resultado é essencial para o entendimento do papel das funções racionais e pode ser facilmente demonstrado. Vamos considerar uma curva projectiva $C/\mathbb{K}$ e uma qualquer função racional $f \in \bar{\mathbb{K}}(C)^*$. Então

154 FACTO

A função f tem um número finito de zeros e de pólos em C e a ordem de cada zero ou pólo é finita. Se f não for uma função constante sobre C (isto é, $f \notin \mathbb{K}^*$) então, pelo menos num ponto $P \in C$ tem-se $\mathrm{ord}_P(f) \neq 0$. *Adicionalmente verifica-se sempre*

$$\sum_{P \in C} \mathrm{ord}_P(f) \;=\; 0$$

Este resultado diz-nos que o número de pólos de f (cada um contanto tantas vezes quanto a sua ordem) tem de ser igual ao número de zeros. Diz-nos também que, a menos do caso trivial das funções constantes sobre na curva[57], existem sempre pólos e zeros e em número finito.

□

Frequentemente interessa-nos resolver o problema inverso:

> *dado um conjunto de pontos eventuais pólos $P_1, P_2, \cdots, P_n$ e de eventuais zeros $Z_1, Z_2, \cdots, Z_n$ quere-se construir uma função racional f que tenha exactamente estes pólos e estes zeros*

Essencialmente quere-se $f = p/q$ definindo dois polinómios p e q homogéneos e com o mesmo grau; o primeiro deve ter os pontos Z_i como raízes e o segundo deve ter os pontos P_i como raízes.

[57] Note-se que f pode ser constante sobre C sem ser uma função constante; por exemplo, se ϕ determinar a curva C, a fracção $(\lambda + \phi)/(\mu + \phi)$, com $\lambda, \mu \in \mathbb{K}^*$, não é constante mas determina uma função racional que é constante sobre C.

Vamos começar por considerar versões simplificadas deste problema começando por polinómios homogéneos lineares; isto é **rectas**. Para isso precisamos de algumas ferramentas que nos ajudem a construir e manipular rectas.

155 DEFINIÇÃO
Sejam $P, Q \in \mathbb{P}^2$ *determinados por representantes* (a, b, c) *e* (a', b', c') *respectivamente. Se* $P \neq Q$ *define-se*

$$P \otimes Q \doteq [bc' - cb' \ , \ ca' - ac' \ , \ ab' - ba'] \tag{138}$$

Representa-se por $\mathrm{l}(P)$ *o polinómio homogéneo do* 1^o *grau* $a\,x + b\,y + c\,z$ *ou, indistintamente, a recta definida por isso polinómio.*

Facilmente se verifica que $P \otimes Q$ e $\mathrm{l}(P)$ são independentes do representante escolhido para os pontos P e Q.

Como $P \otimes$ Q só está definido para pontos distintos, pode-se estender a definição acrescentando um ponto $\mathfrak{O}$ extra ao espaço $\mathbb{P}^2$ (identificado com o triplo de coordenadas todas nulas $(0, 0, 0)$) e fazendo $P \otimes P = P \otimes \mathfrak{O} = \mathfrak{O}$. Também $\mathrm{l}(\mathfrak{O}) \doteq \mathbb{P}^2$. Nestas circunstâncias

156 PROPOSIÇÃO
O operador $\otimes$ *definido em* $\mathbb{P}^2 \cup \{0\}$ *por* (138) *é comutativo. Para* $P, Q \in \mathbb{P}^2$ *verifica-se* $P \otimes Q = \mathfrak{O}$ *se e só se* $P = Q$. *Adicionalmente, para todo* $P, Q, R \in \mathbb{P}^2$, *verifica-se*

$$R \in \mathrm{l}(P \otimes Q) \quad \textit{sse} \quad P \in \mathrm{l}(R \otimes Q) \quad \textit{sse} \quad Q \in \mathrm{l}(P \otimes R)$$

Assim cada ponto determina, através das suas coordenadas, uma recta. A recta que passa pelos pontos P e Q é determinada pelas coordenadas do ponto $P \otimes Q$.

□

Nestas circunstâncias, voltando ao problema inicial de construir funções racionais dados os seus pólos e zeros, tem-se

1. Se pretendermos um polinómio que tenha exactamente dois zeros, P e Q, basta construir a recta $\mathrm{l}(P \otimes Q)$.
2. Se pretender-mos uma função racional que tenha um zero Z e um polo P, podemos começar por escolher um outro qualquer ponto O que seja distinto de Z e de P e construir duas rectas, ambas passando por O, e passando por Z e por P.
$$p = \mathrm{l}(Z \otimes O) \quad , \quad q = \mathrm{l}(P \otimes O)$$
A fracção $f = p/q$ determina a função racional pretendida. Note-se que ela tem um polo em O que se anula com o zero que tem em O; por isso f acaba por ter só o polo P e o zero Z.
3. A estratégia anterior pode ser usada para construir funções racionais com pólos e zeros com ordem superior a 1. Vamos supor que se quer um polo P de ordem 2 e um zero Z também de ordem 2. Então escolhem-se dois quaisquer pontos O_1 e O_2 distintos entre si e distintos de P e Z. Em seguida constroem-se rectas
$$p_1 = \mathrm{l}(Z \otimes O_1) \qquad p_2 = \mathrm{l}(Z \otimes O_2) \qquad q_1 = \mathrm{l}(P \otimes O_1) \qquad q_2 = \mathrm{l}(P \otimes O_2)$$
A fracção $f = (p_1\, p_2) \,/\, (q_1\, q_2)$ determina a função racional pretendida. Note-se que tanto O_1 como O_2 aparecem simultaneamente como zeros e pólos e, por isso, anulam-se.

4. Vamos agora considerar que se quer n zeros $Z_1, \cdots, Z_n$, todos distintos, e n pólos $P_1, \cdots, P_n$ também todos distintos e distintos dos zeros. Basta escolher um ponto auxiliar O e construir as rectas

$$p_i = \mathrm{l}(Z_i \otimes O) \qquad q_i = \mathrm{l}(P_i \otimes O) \qquad i = 1 \cdots n$$

e definir a função racional $f = \prod_{i=1}^{n} (p_i/q_i)$.
Obviamente, se um zero ou um polo aparecer repetido (ordem > 1) temos de usar mais pontos auxiliares tal como fizemos no caso anterior.

Estes casos indicam um algoritmo simples para construir uma função racional dados os seus conjuntos de zeros e pólos. Este algoritmo é particularmente importante em curvas elípticas na construção de emparelhamentos.

8.2 Introdução a ideais e variedades

Problema simportantes, como a caracterização da intersecção de duas curvas, que são fundamentais ao estudo das curvas elípticas requerem uma análise, mesmo resumida, da noção de variedade e, por isso, da noção de ideal.

Seja R um anel; no que se segue vamos sempre assumir que os anéis são formam um domínio integral: isto é, são comutativos e $r, s \neq 0 \in R$ implica sempre $r\,s \neq 0$.

Dados subconjuntos $I, J \subseteq R$ define-se $I+J = \{\, r+s \mid r \in I\ ,\ s \in J \,\}$ e $I\,J = \{\, r\,s \mid r \in I\ ,\ s \in J \,\}$. Define-se, $I^0 = R$ e $I^{n+1} = I\,I^n$. O conjunto $\{s\}J$ escreve-se como sJ.

157 NOÇÃO
Um subconjunto não vazio $I \subseteq R$ é um **ideal** *quando $I\,R = I$ e $I + I = I$.*

Um ideal $\mathfrak{p} \neq R$ é **primo** *quando $r\,s \in \mathfrak{p}$ implica $r \in \mathfrak{p}$ ou $s \in \mathfrak{p}$. O ideal $\mathfrak{m}$ é* **máximo** *quando não está contido em nenhum outro ideal primo.*

Claramente, a soma e o produto de ideais são ideais. Se I é um ideal então o conjunto $\{\, r \mid r^n \in I$ para algum $n \,\}$ é também um ideal. Tal conjunto representa-do por $\sqrt{I}$ e designa-se por **radical** de I. Um **ideal radical** é qualquer ideal I que coincida com o seu radical. Todo o ideal primo $\mathfrak{p}$ é radical.

Cada ideal primo $\mathfrak{p} \subset R$ determina uma relação de equivalência em R da forma usual: $r \sim s$ sse $r - s \in \mathfrak{p}$. Porque $\mathfrak{p}$ é primo, o espaço quociente respectivo tem a estrutura de um domínio integral; representa-se $R/\mathfrak{p}$ tal anel. Seja $I \subset R$ um ideal que contenha $\mathfrak{p}$; define-se $I/\mathfrak{p}$ como o ideal $\mathfrak{q}$ tal que $I = \mathfrak{p} + \mathfrak{q}$. Verifica-se facilmente que $I/\mathfrak{p}$ determina um ideal em $R/\mathfrak{p}$; adicionalmente, todos os ideais em $R/\mathfrak{p}$ têm esta forma.

158 NOÇÃO

Um ideal da forma $s\,R$, com $s \in R$, diz-se **principal** *e representa-se por $\langle s \rangle$. Um ideal I é* **finitamente gerado** *quando se pode escrever como $I = s_1R + s_2R + \cdots + s_nR$, para um conjunto finito $\{s_1, s_2, \ldots, s_n\}$ de elementos de R designados por* **geradores**. *Neste caso escreve-se $I = \langle s_1, s_2, \ldots, s_n \rangle$.*

Se $\mathfrak{p}$ é um ideal primo, o seu **peso** *é o maior k tal que existe uma cadeia $\mathfrak{p}_0 \subset \mathfrak{p}_1 \subset \cdots \subset \mathfrak{p}_{k-1} \subset \mathfrak{p}$ em que os vários $\mathfrak{p}_i$ são ideais primos distintos entre si e distintos de $\mathfrak{p}$. O supremo dos pesos de todos os ideais primos de R designa-se por* **dimensão de Krull** *(ou, simplesmente,* **dimensão***) de R. Um anel de dimensão finita diz-se* **Noeteriano**.

159 TEOREMA (BÁSICO DE HILBERT)

Um anel R é Noeteriano se e só se for finitamente gerado. Adicionalmente, sendo R Noteriano, qualquer anel de polinómios $R[x_1, \cdots, x_n]$ é noeteriano.

□

Considere-se agora um corpo $\mathbb{K}$ e o anel $S_n = \overline{\mathbb{K}}[x_1, \ldots, x_n]$ dos polinómios nas variáveis $x_1, \ldots, x_n$ com coeficientes no fecho algébrico de $\mathbb{K}$. Interessa-nos considerar o anel S_n mas também os anéis quociente

$R = S_n/\mathfrak{p}$, sendo $\mathfrak{p}$ um ideal primo. Qualquer dos casos será designado por um **anel de polinómios**.

Uma observação importante resulta do facto de qualquer corpo $\mathbb{K}$, visto como um anel, ser noeteriano. De facto o único ideal primo de $\mathbb{K}$ é o anel trivial $\{0\}$ que tem peso 1. Então, pelo teorema básico de Hilbert, todo S_n é noeteriano; o que implica

160 COROLÁRIO *Todo o ideal $I \subset S_n$ é finitamente gerado.*

EXEMPLO 37: Tomemos $\mathbb{K}$ como $\mathbb{Q}$ (o corpo dos números racionais) e o anel de polinómios a duas variáveis S_2. Vamos ver alguns exemplos de ideais neste anel. Tenha-se em atenção que $S_2 = \overline{\mathbb{Q}}[x, y]$ e, dado que o fecho algébrico de $\mathbb{Q}$ é o corpo dos complexos $\mathbb{C}$, os elementos de S_2 são polinómios nas variáveis x, y com coeficientes complexos.

Como todo o ideal de S_2 é finitamente gerado (corolário 160) para definir um ideal basta indicar os seus geradores. É possível definir o ideal de outras formas: através de operações sobre outros ideais ou através de definição do conjunto por compreensão a partir de uma propriedade dos polinómios.

Via geradores temos, por exemplo, os ideais

$$I = \langle x, y - x \rangle \qquad J = \left\langle y^2 - x^3 - 1 \right\rangle$$

Pode-se definir ideias através das operações de soma e produto. Por exemplo

$$I + J = \left\langle x, y - x, y^2 - x^3 - 1 \right\rangle \quad , \quad I\,J = \left\langle x\,(y^2 - x^3 - 1), (y - x)\,(y^2 - x^3 - 1) \right\rangle$$

Pode-se finalmente definir ideiais por compreensão: seja $U = \{p_1, \ldots, p_n\}$ um conjunto finito de $\mathbb{K}^2$-pontos, então pode-se definir

$$I_U = \{ f \in S_2 \mid f(p) = 0 \quad \text{para todo } p \in U \}$$

Considere-se, de novo, $S_n = \overline{\mathbb{K}}[x_1, \ldots, x_n]$. Os ideais no anel S_n têm uma relação clara com determinados conjuntos $X \subset \mathbb{K}^n$ designados por **algébricos**. Para todo subconjunto $X \subseteq \mathbb{K}^n$ define-se

$$\mathbf{I}(X) = \{ f \mid f(p) = 0 \text{ para todo } p \in X \} \tag{139}$$

Pode-se verificar que $\mathbf{I}(X)$ é um ideal. Alternativamente seja $I \subset S_n$ um qualquer ideal; define-se

$$\mathbf{Z}(I) = \{ p \in \mathbb{K}^n \mid f(p) = 0 \text{ para todo } f \in I \} \tag{140}$$

Conjuntos $X \subseteq \mathbb{K}^n$ da forma $\mathbf{Z}(I)$, para algum ideal I dizem-se **algébricos**. $\mathbf{Z}$ mapeia ideais em conjuntos algébricos; a construção $\mathbf{I}(X)$ mapeis quaisquer conjuntos em ideais. A relação entre estas duas construções é expressa num dos resultados mais importantes da Álgebra.

161 TEOREMA (NULLSTELLENSATZ)
Todo o ideal $I \subset S_n$ verifica $\mathbf{I}(\mathbf{Z}(I)) = \sqrt{I}$.

Alguns corolários que são consequência imediata deste teorema e do facto do corpo $\overline{\mathbb{K}}$ ser algebricamente fechado.

162 COROLÁRIO *Sejam* $I, J \subset S_n$ *ideais e* $X, Y \subset S_n$ *conjuntos algébricos. Então verifica-se* $\mathbf{Z}(I+J) = \mathbf{Z}(I) \cap \mathbf{Z}(J)$, $\mathbf{Z}(I \cap J) = \mathbf{Z}(I) \cup \mathbf{Z}(J) = \mathbf{Z}(I\,J)$, $\mathbf{I}(X \cap Y) = \sqrt{\mathbf{I}(X) + \mathbf{I}(Y)}$ *e* $\mathbf{I}(X \cup Y) = \mathbf{I}(X) \cap \mathbf{I}(Y)$.

163 COROLÁRIO *Dado um qualquer conjunto* $X \subseteq \mathbb{K}^n$ *(algébrico ou não), seja* $\overline{X}$ *a intersecção de todos os conjuntos algébricos* Y *que contém* X*. Então* $\overline{X} = \mathbf{Z}(\mathbf{I}(X))$.

O conjunto $\overline{X}$ designa-se por **fecho** de X. Se X é algébrico e $X = \overline{X}$, então X diz-se **algébricamente fechado**.

Se existir uma partição $X = Y \cup Y'$, sendo Y e Y' subconjuntos próprios de X que são algébricamente fechados, então X diz-se **redutível**. Se não existir tal partição, X diz-se **irredutível**.

164 COROLÁRIO *Um conjunto algébrico* $X \subset \mathbb{K}^n$ *é irredutível se e só se* $\mathbf{I}(X)$ *é primo. Adicionalmente, se* X *for irredutível, o seu fecho* $\overline{X}$ *também é irredutível.*

165 COROLÁRIO *Um ideal principal* $\langle \phi \rangle$ *é primo se e só se* ϕ *for irredutível em* $\overline{\mathbb{K}}\,[x_1, \ldots, x_n]$.

166 COROLÁRIO *Para todo ponto* $p = (a_1, a_2, \cdots, a_n) \in \mathbb{K}^n$ *, o conjunto singular* $\{p\}$ *é algébrico.*

O ideal $\mathfrak{m}_p = \mathbf{I}(\{p\})$ *é máximo e é gerado pelos* n *polinómios* $\{(x_i - a_i)\}_{i=1}^n$ *. Isto é*

$$\mathfrak{m}_p = \langle\, x_1 - a_1 \;,\; x_2 - a_2 \;,\; \cdots \;,\; x_n - a_n \,\rangle \tag{141}$$

Adicionalmente todo o ideal máximo em S_n *tem a forma* $\mathfrak{m}_p$ *, para algum ponto* $p \in K^n$ *. Finalmente, para todo o ideal* $I \subset R$*, tem-se* $p \in \mathbf{Z}(I)$ *se e só se* $I \subseteq \mathfrak{m}_p$.

Notas Provas para o Nullstellensatz e para estes corolários podem ser obtidas no textos standard de Geometria Algébrica; por exemplo, Commutative Algebra de David Eisenbud ou Algebraic Geometry de Robin Hartshorne, publicados no Graduate Texts in Mathematics da Springer-Verlag.

167 Definição
Uma **variedade afim** *(ou, simplesmente,* **variedade***) é um conjunto algébrico* $V \subseteq \mathbb{K}^n$ *irredutível e algebricamente fechado.*

A **dimensão** *de* V *é o maior* k *para o qual existe uma cadeia* $X_1 \subset X_2 \subset \cdots \subset X_k$, *de subconjuntos próprios* $X_i \subset V$ *que são distintos, irredutíveis e algebricamente fechados. Uma variedade designa-se por* **ponto**, **curva**, **superfície** *ou* **hipersuperfície**, *consoante a sua dimensão é, respectivamente,* 0, 1, 2 *ou* > 2.

O domínio integral $S_n/\mathbf{I}(V)$ *representa-se por* $\mathbb{A}(V)$ *e designa-se por* **anel afim** *de* V.

O conjunto $\mathbb{K}^n$, visto como uma variedade, representa-se por $\mathbb{A}^n$. O respectivo ideal $\mathbf{I}(\mathbb{A}^n)$ é o ideal trivial $\{0\}$. O conjunto vazio, visto como conjunto algébrico, é também uma variedade gerada pelo ideal $\langle 1 \rangle$.

Note-se que, sendo V uma variedade, o corolário 164 diz-nos que $\mathbf{I}(V)$ é um ideal primo e, por isso, o anel quociente $\mathbb{A}(V) = S_n/\mathbf{I}(V)$ é um domínio integral.

Claramente $\mathbb{A}(V)$ tem a estrutura de um espaço vectorial sobre $\overline{\mathbb{K}}$. Anéis que são extensões de um corpo K e são espaços vectoriais sobre esse corpo designam-se por K-**álgebras**. Assim $\mathbb{A}(V)$ é uma $\overline{\mathbb{K}}$-álgebra.

Como espaço vectorial, tem uma dimensão que coincide com o número de elementos numa sua base. No entanto $\mathbf{I}(V)$, como todo o ideal de S_n, é finitamente gerado; isto faz-nos pensar que o espaço vectorial $S_n/\mathbf{I}(V)$ tem uma dimensão finita. De facto, temos um resultado bastante mais forte.

168 Teorema
Seja V uma variedade. O anel afim $\mathbb{A}(V)$ é uma $\overline{\mathbb{K}}$-álgebra finitamente gerada. A dimensão de $\mathbb{A}(V)$ como espaço vectorial coincide com sua dimensão de Krull como anel e coincide também com a dimensão da variedade V.

Uma série de resultados importantes resultam deste teorema.

169 Corolário *A dimensão de $\mathbb{A}^n$ é n. Uma variedade $V \subset \mathbb{A}^n$ tem dimensão $n-1$ se e só se tem $I(V) = \langle \phi \rangle$ para algum polinómio ϕ irredutível em S_n.*

170 Proposição
Seja V uma variedade de dimensão d e seja $\mathfrak{p}$ um ideal primo do anel afim $\mathbb{A}(V)$ de peso p. Então

$$\dim \mathbb{A}(V)/\mathfrak{p} \;=\; d - p$$

□

Os ideias máximos $\mathfrak{m}_p$ e as suas diversas potências $\mathfrak{m}_p^m$, definidos por pontos $p \in K^n$, são essenciais para a definição de multiplicidade (das raízes de um polinómio, da intersecção de duas curvas, etc.). Note-se que, para todo $n \geq 0$, o ideal $\mathfrak{m}_p^n$ está contido em $\mathfrak{m}_p$ e, genericamente, em todos os $\mathfrak{m}_p^k$, com $k < n$.

171 Noção

Seja $\mathfrak{p}$ *um ideal primo e* $p \in K^n$ *um ponto em* $\mathbf{Z}(\mathfrak{p})$ *(isto é, verifica-se* $\mathfrak{p} \subseteq \mathfrak{m}_p$*). Dado um qualquer ideal* $\mathfrak{a} \not\subseteq \mathfrak{p}$*, define-se a* **multiplicidade** *de* $\mathfrak{a}$ *em* p **sobre** $\mathfrak{p}$*, como a maior potência* $m \geq 0$ *tal que* $\mathfrak{a} \subseteq \mathfrak{m}_p^m/\mathfrak{p}$*. Representa-se essa multiplicidade por* $\eta_p(\mathfrak{a};\mathfrak{p})$*.*

Nomeadamente, quando $\mathfrak{p}$ *coincide com o ideal trivial* $\langle 0\rangle$*,* $\eta_p(\mathfrak{a};\langle 0\rangle)$ *representa-se simplesmente por* $\eta_p(\mathfrak{a})$ *e designa-se por* **multiplicidade** *de* $\mathfrak{a}$ *em* p*.*

A situação mais comum verifica-se quando se tem $\mathfrak{p} = \mathbf{I}(V)$, para uma determinada variedade V, e $\mathfrak{a} = \langle f\rangle$ para um polinómio f. Nestas circunstâncias, temos a noção de multiplicidade de um polinómio f num ponto p sobre a variedade V ou, simplesmente, multiplicidade do polinómio f no ponto p.

Exemplo 38: Considere-se o espaço afim $\mathbb{A}^3$ e a variedade afim E definida pelo polinómio $\phi \equiv x^2+y^2+z^2-z$. É facil verificar que E é uma esfera de raio $1/2$ centrada no ponto $(0,0,1/2)$. O plano definido pelo polinómio $p \equiv z$ é tangente à esfera na origem $(0,0,0)$. Trivialmente tem-se

$$z = (x^2+y^2+z^2) - \phi$$

Isto significa que, em $\mathbb{A}(E)$, temos $z \sim (x^2+y^2+z^2)$. O termo $(x^2+y^2+z^2)$ é um factor local na origem $(0,0,0)$ de grau 2; portanto o plano tangente z tem multiplicidade 2 na origem sobre a esfera E.

Numa variedade V, um ponto $p \in V$ diz-se **singular** quando $\eta_p(\mathbf{I}(V)) > 1$. As variedades não-singulares V são aquelas que não têm pontos singulares.

EXEMPLO 39: Tomemos de novo a esfera $E : \phi$, com $\phi \equiv x^2 + y^2 + z^2 - z$, que vimos no exemplo 38. Como determinar a multiplicidade de ϕ num ponto genérico $p = [a, b, c]$ da esfera?

Para isso a melhor solução é usar um sistema de computação algébrica, como o MAPLE, e um pacote de funções, como `PolynomialIdeal`, para computar as várias operações com ideais.

1. Começa-se por determinar o ideal máximo genérico $m_p = \langle x - a, y - b, z - c \rangle$ e as respectivas potências $m_p^2 = m_p\, m_p$, $m_p^3 = m_p^2\, m_p$, etc.
2. Tratando a, b, c como 3 novas variáveis, constrói-se o ideal $S := \langle \phi(a, b, c) \rangle$ que denota o facto de o ponto (a, b, c) pertencer à variedade E. Neste caso tem-se $S = \left\langle a^2 + b^2 + c^2 - c \right\rangle$.
3. No espaço de polinómios a 6 variáveis (x, y, z, a, b, c) constroem-se sucessivamente os ideais $s_1 = m_p + S$, $s_2 = m_p^2 + S$, $s_3 = m_p^3 + S$, etc.
4. Testa-se sucessivamente, $\phi \in s_1$, $\phi \in s_2$, $\phi \in s_3$, etc. O último k onde o teste $\phi \in s_k$ tem sucesso, é a multiplicidade pretendida.

Executando este algoritmo verifica-se que a multiplicidade é sempre 1 e, portanto, a variedade é não-singular.

A noção de ponto singular está assocoada à noção de tangente a uma variedade num ponto e, baseado neste conceito, é possível verificar facilmente se um ponto é ou não singular. Para isso precisamos de uma extensão das noções de jacobiano (noção 181, página 492) e de tangente (noção 149) a variedades.

172 NOÇÃO
Seja $V \subset \mathbb{A}^n$ *a variedade definida pelo ideal* $\langle g_1, \ldots, g_l \rangle$. *O* **jacobiano** *de* V *é a matrix de polinómios* $\mathcal{J}$ *cujo elemento genérico é* $\mathcal{J}_{ij} = \partial g_j / \partial x_i$.

O seguinte teorema[58] permite detectar os eventuais pontos singulares das variedades.

173 TEOREMA
Seja $\mathcal{J}$ *o jacobiano da variedade* V*; um ponto* $p \in V$ *é não-singular se e só*

$$\text{Rank}(\mathcal{J}(p)) \;=\; n - \dim(V)$$

EXEMPLO 40: Considere-se de novo a esfera nos exemplos 38 e 39. Nesta variedade a dimensão é 2 e o espaço tem dimensão 3. Donde, neste caso, $n - \dim(V) = 1$.

O ideal que define a esfera tem apenas um gerador (o polinómio ϕ). Portanto o jacobiano é o vector coluna das derivadas parciais $(\partial\phi/\partial x, \partial\phi/\partial y, \partial\phi/\partial z)$. Neste caso será $\mathcal{J} = (2\,x, 2\,y, 2\,z - 1)$ que, como matriz, tem *rank* 1 para todo ponto da variedade.

[58] Para prova veja-se HARTSHORNE, *Algebraic Geometry*.

Se considerar-mos a intersecção da esfera com o plano $x = 0$ temos um círculo; a dimensão da variedade é 1 donde, neste caso, $n - \dim(V) = 2$.

Os geradores do ideal que define o círculo são $\left\langle x^2 + y^2 + z^2 - z, x \right\rangle$. O jacobiano é a matriz $\begin{bmatrix} 2\,x & 1 \\ 2\,y & 0 \\ 2\,z - 1 & 0 \end{bmatrix}$.

No círculo tem-se $x = 0$. Para que esta matriz tenha *rank* < 2, a primeira coluna terá de ser múltipla da segunda; isto só acontece se for $y = 0$ e $z = 1/2$. No entanto o ponto $(0, 0, 1/2)$ não pertence à variedade. Por isso, neste círculo, o *rank* do jacobiano é sempre 2 e não existem pontos singulares.

Um terceiro exemplo é a variedade de dimensão 2 definida pelo polinómio $\psi = y^2\,z - x\,(x - z)^2$. O jacobiano é a matriz coluna $\mathcal{J} = \left[(x - z)\,(z - 3\,x)\ ,\ 2\,y\,z\ ,\ y^2 + 2\,x\,(x - z)\right]$.

Note-se que qualquer ponto onde seja $x = z$ e $y = 0$ é um ponto da variedade definida por ψ; nesses pontos o jaconiano reduz-se a $\mathcal{J} = [0, 0, 0]$; portanto tem *rank* 0. Como consequência todos os pontos da forma $(x, 0, x)$ são pontos singulares da variedade definida por ψ.

A noção de tangente a uma variedade é um conceito um pouco mais complexo do que o de tangente a uma curva. Por exemplo, em relação a uma superfície $V \subset \mathbb{A}^3$ (como a esfera dos exemplos anteriores) pode-se ver a tangente como um plano, uma recta ou mesmo só um ponto.

Por isso convém estender cuidadosamente este conceito.

Cada ponto $p = (a_1, \ldots, a_n) \in \mathbb{A}^n$ define a variedade $\mathbf{l}(p)$, designada por **hiper-plano** de p, gerada pelo polinómio do 1º grau $a_1\, x_1 + \cdots + a_n\, x_n$. A dimensão de $\mathbf{l}(p)$ é $n-1$, se for $p \neq (0, \ldots, 0)$ ou é n em caso contrário.

Um vector de pontos $L = (p_1, \ldots, p_n) \in (\mathbb{A}^n)^l$ gera a variedade $\mathbf{l}(L) = \bigcap_{j=1}^{l} \mathbf{l}(p_j)$. A dimensão desta variedade depende do número de pontos não nulos que são linearmente independentes. Vendo L como uma matriz, o número de pontos não-nuos linearmente independentes é dado pelo *rank* da matriz. Por isso,

174 FACTO
$\mathbf{l}(L)$ *é uma variedade e um* K*-espaço vactorial de dimensões* $n - \mathrm{Rank}(L)$.

175 NOÇÃO
Dada uma variedade afim $V \subset \mathbb{A}^n$ *de jacobiano* $\mathcal{J}$, *seja* $\mathcal{T}(V)$ *a variedade dada por*

$$\mathcal{T}(V) \;=\; \{\, (p, X) \in V \times \mathbb{A}^n \;\mid\; X \in \mathbf{l}(\mathcal{J}(p)) \,\} \tag{142}$$

Caso V *seja não-singular então* $\mathcal{T}(V)$ *designa-se por* **feixe tangente** *de* V.

Note-se nesta definição que, sendo V é não-singular, temos $\dim(V) = n - \mathrm{Rank}(\mathcal{J}(p)) = \dim(\mathbf{l}(\mathcal{J}(p)))$, independentemente de $p \in V$. Portanto a variedade $\mathbf{l}(\mathcal{J}(p))$ tem, para todo $p \in V$, exactamente a mesma dimensão que V.

8.3 Divisores

Dado que o conjunto de zeros e de pólos caracterizam completamente as funções racionais sobre uma curva C, é conveniente introduzir uma noção que faça esta abstracção; isto é, represente os conjuntos de pólos e zeros com as suas ordens associadas. Esta é a noção de **divisor**.

176 DEFINIÇÃO
Um **divisor** *numa curva projectiva* C *é uma soma formal escrita*

$$D = \sum_{P \in C} n_P (P) \qquad com \quad n_p \in \mathbb{Z} \tag{143}$$

onde o número de inteiros n_P *diferentes de zero é finito.*

A **ordem** *do divisor* D *no ponto* P, *representada* $\mathsf{ord}_P(D)$, *é o inteiro* n_P. *O* **suporte** *de* D, $\mathsf{supp}(D)$, *é o conjunto dos pontos* $P \in C$ *em que* $\mathsf{ord}_P(D) \neq 0$. *A* **grau** *do divisor* (143) *é o inteiro*

$$\mathsf{deg}(D) = \sum_{p \in C} \mathsf{ord}_P(D)$$

O conjunto dos divisores de C *é representado por* Div_C *e o conjunto dos divisores de grau* 0 *(i.e., os que verificam* $(\sum_P n_P) = 0$ *) representa-se por* Div_C^0 .

Um divisor é **efectivo** *(e escreve-se* $D \geq 0$ *) se* $\mathsf{ord}_P(D) \geq 0$ *para todo* P. $D \geq D'$ *é uma abreviatura para* $D - D' \geq 0$.

Se $f \in \mathbb{K}(C)$, *o divisor de* f, *escrito como* (f) *ou* $\mathsf{div}(f)$ *é a soma formal*

$$(f) \doteq \sum_{P \in C} \mathsf{ord}_P(f)\,(P) \tag{144}$$

Divisores D *para os quais existe* $f \in \mathbb{K}(C)$ *tal que* $D = (f)$ *chamam-se* ***divisores principais***.

Como cada $f \in \mathbb{K}(C)$ tem um número finito de pólos e zeros, a soma formal (144) é um divisor.

Todo o divisor D pode ser escrito, de forma única, como $D_0 - D_\infty$ em que D_0 e D_∞ são divisores efectivos de suportes disjuntos. O par de divisores (D_0, D_∞) designa-se por **fraccionamento efectivo** de D.

177 FACTO
O conjunto dos divisores Div_C *determina um grupo abeliano que contém os divisores de ordem zero,* Div_C^0, *como sub-grupo.*

Esboço de prova Pode-se ver o conjunto dos divisores de C, Div_C embebido no espaço vectorial $\mathbb{Z}^C$: cada divisor é um vector de componentes em $\mathbb{Z}$ e com tantas componentes quantos os pontos $P \in C$. Os divisores D são, assim, os elementos de Div_C que têm um

número finito de componentes não nulas. A soma de vectores gera a operação de grupo $+$ nos divisores divisores

$$\sum_{P\in C} n_P P \;+\; \sum_{P\in C} m_P P \;\doteq\; \sum_{P\in C} (n_P + m_p)\, P \tag{145}$$

O elemento neutro é o divisor nulo $\mathcal{N} \doteq \sum_{P\in C} 0\, P$. Com tal soma e elemento neutro, Div_C tem a estrutura de um grupo.

O conjunto dos divisores de grau 0, Div_C^0 , formam um sub-grupo porque, como facilmente se verifica, a soma de dois divisores de grau zero gera de novo um divisor de grau zero.

Os divisores principais (divisores da forma $\mathrm{div}(f)$, com f uma função racional em $\mathbb{K}(C)$) têm um papel especial na teoria dos divisores.

Note-se que f é determinado por fracções homogéneas (onde o grau do numerador é igual ao grau do denominador); por isso é natural pensar-se que o número de zeros do numerador seja igual ao número de zeros do denominador.

A noção de divisor prende-se, obviamente, com a distribuição dos pólos e dos zeros das funções racionais; por isso faz sentido a seguinte definição:

178 DEFINIÇÃO
Para cada divisor $D \in \mathrm{Div}(C)$ *seja*

$$L(D) \;=\; \{0\} \cup \{\, f \in \mathbb{K}(C)^* \;\mid\; D + (f) \geq 0 \,\} \tag{146}$$

$L(D)$ contém, em primeiro lugar e como caso particular, a função constante 0; isto é essencial, como veremos adiante, para a estrutura vectorial que queremos impor a este espaço.

Essencialmente porém, $L(D)$ contém todas as funções racionais não-nulas que têm zeros que "anulam" os pólos de D e que não introduzem pólos adicionais. Note-se que, porque f é racional, o número de pólos de f deve ser igual ao número de zeros de f.

Para percebermos o papel fundamental que estes espaços $L(D)$ têm na construção de curvas, o seguinte exemplo ilustra alguns divisores e a construção do espaço respectivo.

EXEMPLO 41:

1. Vamos supor que se tem $D = (P) + (Q) - (R)$, em que $P, Q, R \in C$ são pontos distintos da curva C.

   Tome-se uma qualquer função racional f candidata pertencer a $L(D)$ com apenas um zero e um pólo; isto é, (f) tem a forma $(A) - (B)$ (com A e B por definir). Temos $D + (f) = (P) + (Q) + (A) - (R) - (B)$ e pretende-se que seja $D + (f) \geq 0$.
   Como nesta soma existem dois pólos que têm de ser anulados e os graus de liberdade são A e B, pode-se escolher $A = R$ e $B = Q$. Isto basta para que $D + (f) = (P) \geq 0$.
   Poderíamos também ter escolhido $B = P$ e obtínhamos $D + (f) = (Q) \geq 0$. Note-se que o valor de A não pode ser alterado.

Seria possível usar um $f \in L(D)$ que tivesse dois pólos e dois zeros?

$$(f) \;=\; (A) + (A') - (B) - (B')$$

Neste caso os 4 pontos A, A', B, B' têm de ser distintos e será

$$D + (f) \;=\; (P) + (Q) + (A) + (A') - (R) - (B) - (B')$$

Pode-se usar P e Q para anular B e B' e usar A ou A' para anular R.

Este é o caso limite: não existe nenhum $f \in L(D)$ com três zeros e três pólos porque D só tem 2 zeros para anular os pólos de f.

2. $D = n(P) - m(Q)$, com $n, m > 0$ e $P \neq Q$.

Considere-se uma função racional f tal que $(f) = k(A) - k(B)$. Note-se que o número de zeros tem de ser igual ao número de pólos.

Neste caso, $D + (f) = n(P) + k(A) - m(Q) - k(B)$; para este divisor ser efectivo, terá de ser

$$A = Q \quad \text{e} \quad k \geq m \quad \text{e} \quad B = P \quad \text{e} \quad k \leq n$$

Se for $n = m$ (isto é, se D tiver grau zero), então existe uma única possibilidade: f tem de ter o divisor $n(Q) - n(P) = -D$.

179 LEMA *Seja D um divisor de uma curva projectiva C e $L(D) \doteq \{ f \in \mathbb{K}(C)^* \mid D + (f) \geq 0 \} \cup \{0\}$. Então $L(D)$ é um espaço vectorial sobre $\bar{\mathbb{K}}$ cuja dimensão, denotada por $\ell(D)$, é finita.*

Prova Afirmar que $L(D)$ é um espaço vectorial é equivalente a dizer que, para todos $f, g \in L(D)$ e $a \in \bar{K}^*$, se verifica $f+g \in L(D)$ e $a\,f \in L(D)$.

Como af ou é zero (se $a = 0$) ou, se $a \neq 0$, tem os mesmos zeros e pólos que f, então $f \in L(D)$ implica $af \in L(D)$. Do mesmo modo, para todo ponto $P \in C$, a ordem $\text{ord}_P(f+g)$ é sempre maior ou igual que $\text{ord}_P(f)$ e $\text{ord}_P(g)$. Por isso, $f, g \in L(D)$ implica $(f+g) \in L(D)$.

Provar que a dimensão do espaço vectorial é finita é mais complexo. No entanto basta recordar que, se fosse infinita, seria possível construir uma combinação linear infinita de funções racionais linearmente independentes.

180 FACTO
Se $D' = D + (h)$, para algum $h \in \mathbb{K}(C)^*$, então os espaços vectoriais $L(D)$ e $L(D')$ são isomórficos.

Prova Considere-se o morfismo $f \mapsto h\,f$ entre os espaços vectoriais $L(D')$ e $L(D)$. Como $D' = D + (h)$ então $f \in L(D')$, se não for 0, se e só se $(f) + (h) + D = (fh) + (D) \geq 0$. Portanto $f \in L(D')$ se e só se $fh \in L(D)$.

Este resultado justifica a seguinte definição

181 Definição
Dois divisores D, D' *numa curva projectiva* C *são equivalentes, e escreve-se* $D \sim D'$, *se existir* $h \in \mathbb{K}(C)$ *tal que* $D - D' = (h)$.

Esta relação (que facilmente se verifica ser uma relação de equivalência) induz um espaço quociente $\mathrm{Div}^0_C/\sim$ no conjunto de divisores de grau zero que vai ter um papel fundamental na construção das curvas elípticas. Como claramente $\mathrm{Div}^0_C/\sim$ herda de Div^0_C a estrutura do grupo abeliano, designa-se este espaço por **grupo de Picard** , normalmente representado por Pic_C .

O teorema que permite relacionar a estrutura de grupo das curvas abelianas com divisores.

182 Teorema (Riemann-Roch)
Dada uma curva projectiva absolutamente irredutível C *existe uma constante* $g \geq 0$ *tal que, para todo o divisor* $D \in \mathrm{Div}(C)$,

$$\ell(D) \geq \deg(D) + 1 - g$$

Adicionalmente, se $2g \leq \deg(D) + 1$, *verifica-se*

$$\ell(D) = \deg(D) + 1 - g \tag{147}$$

Prova A prova deste teorema requer noções que saem fora do âmbito deste trabalho. A estrutura essencial da prova (na sua forma mais recente) pode ser vista no artigo

* http://planetmath.org/encyclopedia/ProofOfRiemannRochTheorem.html.
Uma breve história do teorema e da sua importância pode ser vista em
* http://en.wikipedia.org/wiki/Riemann-Roch_Theorem.

A constante g é um invariante da curva C e designa-se por **genus** da curva. A informação essencial que resulta deste teorema é que g é independente do divisor D e a igualdade (147) verifica-se para todo o divisor cujo grau seja maior ou igual que $2\,g-1$.

Algumas consequência imediatas do teorema de Riemann-Roch

183 PROPOSIÇÃO
Nas condições do teorema 182.

(1) Se C tem genus $g=0$ então existem pontos distintos $P\neq Q\in C$ tais que $(P)\sim(Q)$.

(2) Se C tem genus $g=1$ verifica-se $(P)\sim(Q)$ se e só se $P=Q$.

Prova Sejam P,Q dois pontos tais que $(P)\sim(Q)$ e seja $h\in\bar{\mathbb{K}}(C)$ tal que $(P)=(Q)+(h)$. Daqui conclui-se que $h\in L((Q))$ e, caso seja $P\neq Q$, o morfismo $f\mapsto hf$ estabelece um isomorfismo entre $L((P))$ e $L((Q))$ (ver lema 179). Como $(P)\geq 0$ e $(Q)\geq 0$, ambos os espaços $L((P))$ e $L((Q))$ contém a função constante 1 (já que 1 não tem zeros nem pólos).

Ambos os divisores (P) e (Q) têm grau 1. Se a curva tem genus 0, então $\deg((P))+1\geq 2g$ e a dimensão $\ell((P))=\ell((Q))$ é, pelo teorema de Riemann-Roch, igual a 2. Neste caso é possível existir $h\neq 1$ que mapeia a função $1\in L((P))$ na função $h\in L((Q))$; portanto pode ser $P\neq Q$.

Caso a curva tenha genus 1, já a dimensão $\ell((P)) = \ell((Q)) = 1$ e, por isso, tanto $L((P))$ como $L((Q))$ não podem conter outras funções que não sejam constantes; por isso h tem de ser constante e daí só pode ser $P = Q$.

184 PROPOSIÇÃO

Nas condições do teorema 182, se C tem genus 1 e um ponto $\mathcal{O}$ então existe um morfismo $\sigma : \mathrm{Div}_C^0 \longrightarrow C$ que para cada divisor $D \in \mathrm{Div}_C^0$, existe um único ponto da curva $\sigma(D)$ tal que $D \sim (\sigma(D)) - (\mathcal{O})$. Adicionalmente

(i) σ induz um isomorfismo entre o grupo de Picard Pic_C e a curva C.

(ii) $D \in \mathrm{Div}_C^0$ é principal se e só se $\sigma(D) = \mathcal{O}$.

Prova

(1) Seja D um qualquer divisor de grau 0; então $D + (\mathcal{O})$ tem grau 1 e, numa curva de genus 1, o teorema de Riemann-Roch diz-nos que $\ell(D + (\mathcal{O})) = 1$.
Seja f um gerador de $L(D + (\mathcal{O}))$. Então será simultaneamente, $\deg((f)) = 0$, porque f é racional, $\deg(D) = 0$, por hipótese, e $(f) + D + (\mathcal{O}) \geq 0$ porque $f \in L(D + (\mathcal{O}))$.
O único modo de compatibilizar estas três relações é existir um P tal que

$$(f) + D + (\mathcal{O}) \;=\; (P)$$

ou seja $D + (f) = (P) - (\mathcal{O})$ e, portanto

$$(P) - (\mathcal{O}) \;\sim\; D \tag{148}$$

Este P é único. De facto se tivermos $D \sim D'$ e $D' \sim (P') - (\mathcal{O})$, teríamos necessariamente

$$(P) - (\mathcal{O}) \sim (P') - (\mathcal{O}) \;\Rightarrow\; (P) \sim (P')$$

e, como acurva tem genus 1, tal implica (como acabámos de ver) $P = P'$.

Portanto fica bem definido uma função $\sigma : \mathrm{Div}_0(C) \to C$ que mapeia D no ponto P que verifica (148).

(2) Como consequência adicional vemos que esta construção associa dois divisores equivalentes exactamente ao mesmo ponto. Ou seja, $D \sim D'$ implica $\sigma(D) = \sigma(D')$. Por isso fica definido uma função que mapeia classes de equivalência de divisores em pontos da curva

$$\tilde{\sigma}\colon \mathrm{Pic}(C) \to C \qquad \tilde{\sigma}([D]) = \sigma(D)$$

Esta função tem uma inversa óbvia: a aplicação $\tilde{\sigma}^{-1}\colon P \mapsto [(P) - (\mathcal{O})]$ que associa um ponto $P \in C$ à classe de equivalência do divisor $(P) - (\mathcal{O})$. Assim, $\tilde{\sigma}$ é bijectiva.

(3) Se $\sigma(D) = O$ então $D \sim (\mathcal{O}) - (\mathcal{O})$. Isto significa que, para algum $f \in K(C)$,

$$D = (f) + (\mathcal{O}) - (\mathcal{O}) = (f)$$

Inversamente, se $D = (f)$, então $D + (f) = (\mathcal{O}) - (\mathcal{O})$ o que implica $D \sim (\mathcal{O}) - (\mathcal{O})$ e, por isso, $\sigma(D) = \mathcal{O}$.

185 DEFINIÇÃO

Seja C uma curva projectiva de genus 1 onde está identificado um ponto $\mathcal{O}$. Seja $\sigma\colon \mathrm{Div}^0_C \to C$ definido na proposição 184. Sejam $P, Q \in C$ pontos arbitrários da curva e $n \in \mathbb{Z}$ um inteiro arbitrário. Define-se

$$\begin{array}{lll} P \oplus Q = \sigma((P) + (Q) - 2(\mathcal{O})) & & -P = \sigma((\mathcal{O}) - (P)) \\ [n]P = \sigma(n(P) - n(\mathcal{O})) & [0]P = \mathcal{O} & [-n]P = \sigma(n(\mathcal{O}) - n(P)) \end{array}$$

186 PROPOSIÇÃO

Nas condições da definição 185,

(i) $\langle C, \oplus, O\rangle$ *tem a estrutura de um grupo abeliano e o isomorfismo* $\mathrm{Pic}_C \tilde{\to} C$ *preserva a estrutura de grupos.*

(ii) Seja $D = \sum_{P\in C} n_P(P)$, *um divisor de grau* 0 *arbitrário; então* $\sigma(D) = \bigoplus_{P\in C} [n_P]P$.

Prova

(i) Vimos que $\tilde{\sigma}$ é uma bijecção. Esta função transforma-se num homomorfismo de grupos abelianos se, em C, se optar pela a estrutura mapeada por $\tilde{\sigma}$ a partir das operações de grupo de $\mathrm{Pic}(C)$. Isto é o que ocorre quando se define

$$P \oplus Q = \sigma((P) + (Q) - 2(\mathcal{O})) = \sigma((P) - (\mathcal{O}) + (Q) - (\mathcal{O})) = \sigma(\sigma^{-1}(P) + \sigma^{-1}(Q))$$

(ii) Note-se que a definição de $[n]P$ é equivalente a $([n]P) - (\mathcal{O}) \sim n(P) - n(\mathcal{O})$. Seja $Q \doteq \bigoplus_{P\in C} [n_P]P$. Pela definição da operação $\oplus$ temos

$$\begin{aligned}(Q) - (\mathcal{O}) \sim & \sum \left(([n_P]P) - (\mathcal{O})\right) \\ \sim & \sum n_P(P) - \sum n_P(\mathcal{O}) = D - \left(\sum n_p\right)(\mathcal{O}) = \\ = & D - 0(\mathcal{O}) \;=\; D\end{aligned}$$

Portanto temos $D \sim (Q) - (\mathcal{O})$ o que significa que $Q = \sigma(D)$.

Como consequência imediata de (ii) temos

187 Corolário *Seja C uma curva elíptica e $D = \sum_{P \in C} n_P (P)$ um qualquer divisor de grau 0. Então*

$$D \sim 0 \quad \text{sse} \quad \bigoplus_{P \in C} [n_P]P = \mathcal{O}$$

Vimos que uma função racional $f \in K(C)$ determina uma aplicação dos pontos da curva em $\bar{K}$: para cada ponto P é bem definido o valor $f(P)$.

Faz sentido tentar estender este conceito para divisores. Note-se que divisores representam, essencialmente, arranjos de zeros e de pólos de funções racionais; por isso faz sentido pensar que uma soma formal de zeros e pólos associamos o produto dos valores da função nestes pontos.

188 Noção
Seja C uma curva elíptica sobre K e $h \in K(C)$ uma qualquer função racional sobre essa curva. Para todo o divisor de grau zero $D = \sum_{P \in C} n_P (P)$ define-se

$$h(D) = \prod_{P \in C} h(P)^{n_P} \tag{149}$$

A valoração de uma função racional homogénea h num divisor de grau 0 mantém-se invariante se a função h, nos pontos da curva, sofrer uma "mudança de escala". Isto é,

189 FACTO

Sejam $h, h' \in K(C)$ funções tais que, para alguma constante $\lambda \in K^*$, verifica-se $h'(P) = \lambda\, h(P)$, para todo o ponto $P \in C$. Então $h'(D) = h(D)$ para todo o divisor D de grau 0.

Prova Seja $D_0 = \sum_i n_i(P_i)$ e $D_\infty = \sum_j m_j\,(Q_j)$ (com $n_i, m_j > 0$) o fraccionamento efectivo de D. Seja k o grau destes divisores; isto é $k = \sum_i n_i = \sum_j m_j$.

$$h'(D) = \frac{\prod_i \lambda^{n_i}\, h(P_i)^{n_i}}{\prod_j \lambda^{m_j}\, h(Q_j)^{m_j}} = \frac{\lambda^{\sum_i n_i}\, \prod_i h(P_i)^{n_i}}{\lambda^{\sum_j m_j}\, \prod_j h(Q_j)^{m_j}} = \frac{\lambda^k\, \prod_i h(P_i)^{n_i}}{\lambda^k\, \prod_j h(Q_j)^{m_j}} = h(D)$$

O seguinte teorema é uma caracterização fundamental da valoração de funções racionais em divisores principais.

190 TEOREMA (RECIPROCIDADE DE WEIL)

Seja C uma curva elíptica sobre K. Sejam $f, g \in K(C)$ funções racionais sobre C. Se o suporte de (f) e de (g) forem disjuntos, então

$$f((g)) = g((f))$$

□

Numa curva projectiva C de genus 1 onde está identificado um ponto específico O, são importantes os diferentes

divisores $n(\mathcal{O})$ com $n = 1, 2, \cdots$. Seja

$$L_n \;=\; L(n(\mathcal{O})) \;=\; \{\, f \in K(C) \;\mid\; (f) + n(\mathcal{O}) \geq 0 \,\} \cup \{0\}$$

Pode-se ver L_n como o espaço das funções racionais de C que não têm qualquer pólo a não ser em $\mathcal{O}$ e aí a ordem não é superior a n. Note-se que estes espaços vectoriais formam, por inclusão, uma cadeia ascendente

$$L_1 \subset L_2 \subset \cdots \subset L_n \subset \cdots$$

Como a curva tem genus 1, a dimensão de L_n é n (pelo teorema de Rieman-Roch). Portanto o espaço L_n tem n geradores linearmente independentes. Quais são estes geradores?

O espaço L_1 contém todas as funções racionais constantes e, como a sua dimensão é 1, estas definem todos os seus elementos. Deste modo não pode haver nenhuma função racional $f \in K(C)$ que tenha apenas um pólo simples em O: ou não tem qualquer pólo (e é uma constante) ou contém pelo menos outro pólo.

Portanto L_1 tem, por gerador a função unidade 1 : isto é,

$$L_1 \;=\; \{\, a\,1 \;\mid\; a \in K \,\}$$

L_2 tem dimensão 2 e contém L_1; portanto vai existir um gerador não constante g tal que

$$L_2 \;=\; \{\, a\,1 + b\,g \;\mid\; a, b \in K \,\}$$

Note-se, pelo que foi dito antes, que g tem um pólo único de ordem 2 em $\mathcal{O}$.

Considere-se, agora, L_3. Como contém L_2 e tem dimensão 3 será da forma

$$L_3 \;=\; \{\, a\,1 + b\,g + c\,h \;\mid\; a, b, c \in K \,\}$$

em que o novo gerador h é uma função racional com um pólo único de ordem 3 em $\mathcal{O}$.

Os geradores de L_3, $\{1, g, h\}$, são linearmente independentes uma vez que têm ordens dos pólos em $\mathcal{O}$ diferentes. Como consequência, este triplo define uma aplicação racional

$$[g, h, 1] \colon C \setminus \{O\} \to \mathbb{P}^2 \tag{150}$$

entre curva C e o espaço projectivo $\mathbb{P}^2$. Como g e h não têm pólos em C, para além de $\mathcal{O}$, esta aplicação é definida para todo $P \in C \setminus \{O\}$.

Escolham-se polinómios homogéneos do mesmo grau $u, v, w \in K[C]$ tais que w tem um zero triplo em $\mathcal{O}$ e mais nenhum zero em pontos de C, v/w é um representante de h e u/w é um representante de g. É possível escolher tais polinómios porque, em C, h e g só têm pólos em $\mathcal{O}$, sendo triplo o pólo de h e duplo o pólo de g. Assim, em $\mathcal{O}$, u tem de ter um zero simples (já que o pólo de g em $\mathcal{O}$ tem ordem 2) e v não tem nenhum zero.

Define-se agora $\varphi\colon C \longrightarrow \mathbb{P}^2$ por este triplo de polinómios

$$\varphi = [u, v, w] \quad \text{com} \quad g = [u/w] \ , \ h = [v/w] \tag{151}$$

191 LEMA *φ definido em* (151) *é um morfismo injectivo* $\varphi\colon C \longrightarrow \mathbb{P}^2$ *tal que* $\varphi(\mathcal{O}) = P_\infty$ *e, para* $P \neq \mathcal{O}$, $\varphi(P) = [g(P), h(P), 1]$.

Prova Se $P = \mathcal{O}$ tem-se $\varphi(P) = [u(\mathcal{O}), v(\mathcal{O}), w(\mathcal{O})] = [0, v(\mathcal{O}), 0] = [0, 1, 0] = P_\infty$. Se $P \neq \mathcal{O}$ tem-se $w(P) \neq \mathcal{O}$ e, por isso, $\varphi(P) = [u(P), v(P), w(P)] = [u(P)/w(P), v(P)/w(P), 1] = [g(P), h(P), 1]$. Resta provar que ϕ é injectivo. Vamos supor que existiam dois pontos $P \neq Q$ tais que $\varphi(P) = \varphi(Q)$. Então necessariamente, para qualquer $f \in L_3$, seria $f(P) = f(Q)$ já que f é uma combinação linear de g e h. Tome-se agora um qualquer ponto $A \in C$ distinto de O, P ou Q e considere-se duas funções distintas $u, v \in L_3$ que partilhem o mesmo zero A. Então, como $u(P) = u(Q)$ e $v(P) = v(Q)$, a função racional $(u - v)$ pertence a L_3 e tem zeros em A, P e Q; logo $(A) + (P) + (Q) \sim 3(\mathcal{O})$; como A é arbitrário , isto não é possível. Consequentemente, φ é um morfismo injectivo.

O seguinte lema estabelece uma relação particular entre divisores de C efectivos de ordem 3 e rectas em $\mathbb{P}^2$.

192 LEMA *Seja* $D \geq 0$ *um divisor efectivo sobre* C *de grau* 3 *que verifica* $D \sim 3(\mathcal{O})$. *Então* φ, *definido em* (151), *mapeia todos os pontos de* C *de ordem não nula em* D *sobre uma mesma recta de* $\mathbb{P}^2$.

Prova Seja f tal que $D = (f) + 3(\mathcal{O})$; porque $D \geq 0$, a função f é um elemento de $L_3 = L(3(\mathcal{O}))$. Então f pode-se escrever como uma combinação linear dos geradores $\{1, g, h\}$; isto é, $f = ag + bh + c$ para alguns $a, b, c \in K$.

Seja $\mathbf{l}$ a recta em $\mathbb{P}^2$ determinada pelo polinómio $aX + bY + cZ$. Como, para todo $P \in C$, se tem $\mathbf{l}(\varphi(P)) = f(P)$, concluímos que P é um zero de f se e só se $\varphi(P)$ for um ponto da recta $\mathbf{l}$. Mas os zeros de f são precisamente os pontos de C que têm ordem não nula em D. Portanto todos os pontos de ordem não nula de D são pontos da recta $\mathbf{l}$.

A partir dos geradores $\{1, g, h\}$ de L_3 constrói-se os geradores dos espaços seguintes. Por exemplo, g^2 tem um pólo único de ordem 4 em $K(C)$; mais nenhum polinómio construído com estas 3 funções tais pólos. Logo os geradores de L_4 são $\left\{1, g, h, g^2\right\}$.

Da mesma forma se conclui que $g\,h$ tem um pólo único de ordem 5 em $\mathcal{O}$ e mais nenhuma combinação polinomial das três funções tem tais pólos. Logo os geradores de L_5 são $\left\{1, g, h, g^2, g\,h\right\}$.

Quando se chega a L_6, porém, algo de novo ocorre. Pode-se construir o pólo de ordem 6 de dois modos diferentes: ou com g^3 ou, então, com h^2. Desta forma existem 7 candidatos a geradores, $\left\{1, g, h, gh, g^2, g^3, h^2\right\}$. Como o espaço só tem dimensão 6, as 7 funções não podem ser linearmente independentes. Como as 5 primeiras têm de pertencer à base de geradores (porque são geradores de L_5), então h^2 (ou g^3) devem ser representáveis como combinação linear dos restantes 6 elementos.

Por isso tem de existir coeficientes $c_i \in K$ tais que

$$h^2 = g^3 + c_4 g\,h + c_3 g^2 + c_2 h + c_1 g + c_0 \tag{152}$$

O coeficiente de g^3 tem de ser $\neq 0$ porque não seria possível, de outra forma, que esta igualdade se verificasse e que h^2 tivesse o pólo de ordem 6 no ponto $\mathcal{O}$; sendo assim, pode-se assumir que o coeficiente é 1.

Tradicionalmente esta equação escreve-se (usando uma outra sequência de coeficientes) de forma algo diferente

$$h^2 + a_1 g\, h + a_3 h = g^3 + a_2 g^2 + a_4 g + a_6 \tag{153}$$

que se designa por **forma de Weierstraß**. Isto permite-nos enunciar um segundo lema

193 LEMA *Seja E a curva em $\mathbb{P}^2$ determinada pelo polinómio*

$$\phi = Y^2\, Z + a_1\, X\, Y\, Z + a_3\, Y\, Z^2 - X^3 - a_2\, X^2\, Z - a_4\, X\, Z^2 - a_6\, Z^3 \tag{154}$$

E é uma curva não-singular e φ definido no lema 191 é um isomorfismo entre C e E

Prova Substituindo h por v/w e g por u/w a igualdade (153) pode-se escrever

$$v^2\, w + a_1\, u\, v\, w + a_3\, v\, w^2 - u^3 - a_2\, u^2 w - a_4\, u\, w^2 - a_6\, w^3 \;=\; 0$$

Isto é equivalente à afirmação de que $\phi(\varphi(P)) = 0$ para todo o ponto P da curva; portanto φ, definido no lema 191, é um morfismo de C para E. O referido lema diz-nos que φ é um isomorfismo entre C e a sua imagem; como é surjectivo, concluímos que a imagem tem de coincidir com E e que φ é um isomorfismo entre C e E.

A consequência essencial destes lemas é

194 TEOREMA
Qualquer curva elíptica C/K sobre o espaço projectivo $\mathbb{P}^2$ é isomórfica com uma curva plana não singular determinada por uma polinómio $\phi \in \mathbb{K}[X, Y, Z]$ da forma (154).

No que se segure C/K é uma curva de genus 1 em $\mathbb{P}^2$ com ponto identificado $\mathcal{O}$.

195 COROLÁRIO *As funções racionais $g = x/z$ e $h = y/z$ são geradores g e h de L_3.*

Como consequência do lema 192 tem-se ainda,

196 LEMA *Seja $\langle C, \oplus, \mathcal{O}\rangle$ a estrutura de grubo abeliano definido em C (definição 185). Então os pontos P, Q e R de C são mapeados por φ em pontos sobre uma mesma recta de $\mathbb{P}^2$ se e só se $-R = P \oplus Q$.*

Prova Seja D o divisor $(P)+(Q)+(R)$ que é efectivo e tem grau 3. Pela definição das operações de grupo tem-se $(P)+(Q)-2(\mathcal{O}) \sim (\mathcal{O})-(R)$ o que é equivalente a ser $D = (P)+(Q)+(R) \sim 3(\mathcal{O})$. Pelo lema 192 tem-se $D \sim 3(\mathcal{O})$ se e só se os pontos P, Q e R são mapeados em pontos sobre uma mesma recta de $\mathbb{P}^2$.

O seguinte lema é verdadeiramente "multiusos"

197 LEMA *Sejam P, Q, R, S pontos de C tais que tanto P como Q são distintos de R ou S e tais que $P \oplus Q = R \oplus S$.*

(i) *Se* $P \neq Q$ *seja* $p = \mathbb{l}(P \otimes Q)$ *a recta que passa pelos pontos* P *e* Q*. Se for* $P = Q$ *seja* $p = \mathbb{l}(\mathcal{J}^C(P))$ *a recta tangente à curva em* P*.*

(ii) *Se* $R \neq S$ *seja* $q = \mathbb{l}(R \otimes S)$ *a recta que passa pelos pontos* R *e* S*. Se for* $R = S$ *seja* $q = \mathbb{l}(\mathcal{J}^C(R))$ *a recta tangente à curva em* R*.*

Então $(P) + (Q) = (R) + (S) + (p/q)$*.*

Prova

1. No caso em que todos os pontos são distintos. Pelo lema 192 a recta $p = \mathbb{l}(P \otimes Q)$ verifica $(p) = (P)+(Q)+(-(P \oplus Q))-3(\mathcal{O})$. Pela mesma razão a recta $q = \mathbb{l}(R \otimes S)$ verifica $(q) = (R) + (S) + (-(R \oplus S)) - 3(\mathcal{O})$. Dado que, por hipótese, se tem $P \oplus Q = R \oplus S$ conclui-se que $(p/q) = (p) - (q) = (P) + (Q) - (R) - (S)$.
2. Se $P = Q$, e pelo mesmo motivo, a recta p verifica $(p) = 2(P) + (-(P \oplus P)) - 3(\mathcal{O})$. O resto da prova é idêntica.

Quando $S = P \oplus Q$, $R = \mathcal{O}$ a função racional p/q é determinada por P e Q (a menos da multiplicação por uma constante) e faz sentido representá-la por um símbolo apropriado. Assim

198 NOÇÃO

Dados pontos P, Q *numa curva elíptica* C *com ponto no infinito* $\mathcal{O}$*; Define-se* $\mu(P, Q) = p/q$ *em que o par de rectas* p *e* q *é dado por:*

(i) *Se* $P \neq Q$ *então* $p = \mathbf{l}(P \otimes Q)$ *é a recta que passa pelos pontos* p *e* Q.

(ii) *Se* $P = Q$, *então* $p = \mathbf{l}(\mathcal{J}^C(P))$ *é a recta tangente à curva em* P.

(iii) *Se* $P \neq -Q$ *então* $q = \mathbf{l}((P \oplus Q) \otimes \mathcal{O})$.

(iv) *Se* $P \oplus Q = \mathcal{O}$ *então* $q = \mathbf{l}(\mathcal{J}^C(\mathcal{O}))$ *é a recta tangente à curva em* $\mathcal{O}$.

199 PROPOSIÇÃO
Para todo $P, Q \in C$ *verifica-se* $(\mu(P, Q)) = (P) + (Q) - (P \oplus Q) - (\mathcal{O})$.

Prova Consequência imediata do lema 197.

EXEMPLO 42: Considere-se uma curva elíptica sobre o corpo $\mathbb{Q}$ aqui determinada pela sua parte afim

$$E\colon y^2 = x^3 - x + 1$$

Nessa curva tomemos por referência um "ponto central" R de coordenadas afins $(0, 1)$.

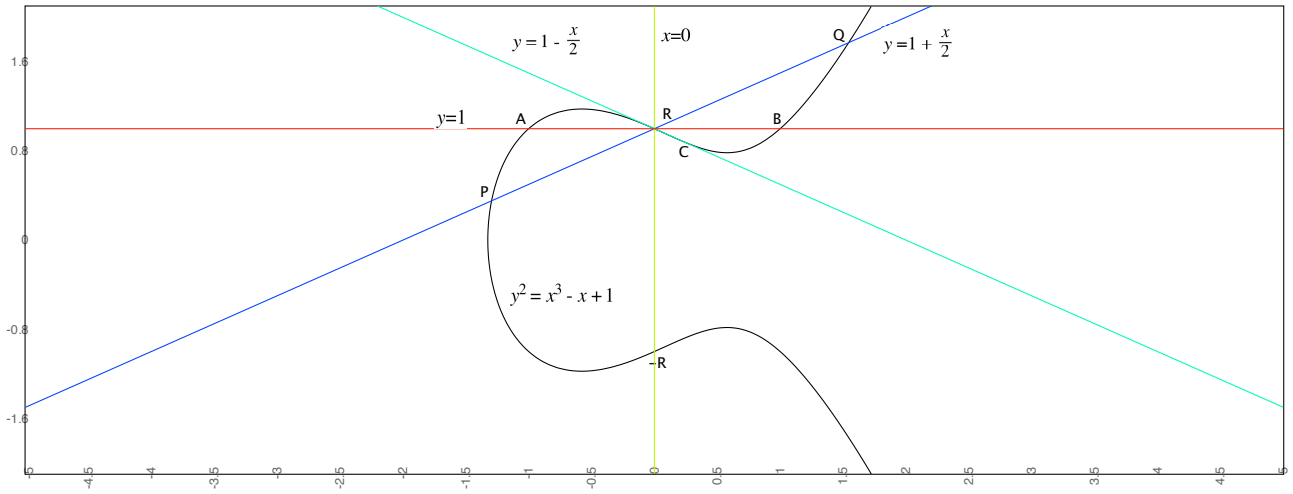


Figura 13: Rectas sobre uma curva elíptica

Vamos considerar a família das rectas que passam por este pontos, como se ilustra na figura 13. Contando com o ponto no infinito P_∞ cada uma das rectas "intersecta" a curva em mais outros 2 pontos.

Na figuras estão indicadas 4 dessas rectas:

1. Uma recta $y = 1 + \frac{x}{2}$ que intersecta a curva em outros dois pontos que designaremos por P e Q.

A recta é determinada pelo polinómio homogéneo

$$x - 2y + 2z$$

que, como elemento de $K[C]$, tem zeros precisamente nesses três pontos. Portanto

$$(x - 2y + 2z) \;=\; (P) + (R) + (Q)$$

Resolvendo o sistema de equações definido pela recta e pela equação da curva pode-se calcular as coordenadas de P e Q. Verifica-se facilmente que

$$P = [\frac{1 - \sqrt{129}}{8} \;,\; \frac{15 + \sqrt{129}}{16} \;,\; 1] \qquad Q = [\frac{1 + \sqrt{129}}{8} \;,\; \frac{17 + \sqrt{129}}{16} \;,\; 1]$$

2. Uma recta $y = 1$ que intersecta a curva em outros dois pontos A e B.
   Em coordenada projectivas o polinómio homogéneo definidor da recta é

$$y - z$$

e os pontos A e B têm coordenadas $[-1, 1, 1]$ e $[1, 1, 1]$ respectivamente. Neste caso temos

$$(y - z) \;=\; (A) + (R) + (B)$$

3. Uma recta $x = 0$ que intersecta a curva (para além de R) no ponto $-R = [0, -1, 1]$ e no ponto do infinito $P_\infty = [0, 1, 0]$.
   O polinómio homogéneo que define esta recta é simplesmente x. Por isso tem-se

$$(x) \;=\; (R) + (-R) + (P_\infty)$$

4. Finalmente temos uma recta $y = 1 - \frac{x}{2}$ que é tangente à curva no ponto R.
   Isto equivale a dizer que o polinómio homogéneo correspondente

$$x + 2y - 2z$$

   tem um zero duplo em R. Resolvendo o sistema de equações verifica-se que tem um terceiro zero no ponto C de coordenadas $[1/4, 7/8, 1]$. Por isso

$$(x + 2y - 2z) \;=\; 2(R) + (C)$$

A soma dos três pontos de uma curva elíptica que estão sobre uma mesma recta é sempre P_∞. Em cada uma destas 4 rectas, um dos pontos é sempre R; logo, a soma dos dois restantes tem de ser constante e igual a $-R$.

$$P \oplus Q = A \oplus B = R \oplus C = -R \oplus P_\infty = -R$$

Combinando as rectas duas a duas construímos várias funções racionais e determinamos os respectivos divisores; por

exemplo

$$\left(\frac{x-2y+2z}{y-z}\right) = (P)+(Q)-(A)-(B)$$

$$\left(\frac{x-2y+2z}{x}\right) = (P)+(Q)-(-R)-(P_\infty)$$

$$\left(\frac{x+2y-2z}{y-z}\right) = (R)+(C)-(A)-(B)$$

etc $\cdots$

8.4 Isogenias e Grupos de Torsão

Vamos continuar a considerar curvas projectivas de genus 1 com um ponto $\mathcal{O}$ definidas no espaço projectivo $\mathbb{P}^2$ gerado por um corpo finito K. Uma tal curva genérica, que designamos por **curva elíptica**, é representada por E/K.

No que se segue vamos sempre assumir que o corpo K é finito e, por isso, o número de pontos de E/K, representado por $|E/K|$ é finito.

Vimos que estas curvas estava definida uma estrutura de um grupo abeliano $\langle E/K, \oplus, \mathcal{O}\rangle$, com a operação do grupo apresentada na definição 185 (ver página 495). Temos, por isso, um grupo abeliano finito.

O objectivo do uso de curvas elípticas em Criptografia reside na possibilidade de construção de grupos cíclicos que sejam sub-grupos deste grupo finito. Para isso vamos considerar o **produto escalar** também apresentado na definição atrás referida.

Nessa mesma definição, para cada $n \in \mathbb{Z}$, definimos uma função $[n]\colon E/K \to E/K$ que mapeia um ponto genérico $P \in E/K$ num ponto $[n]\,P$.

Não é difícil verificar que esta função preserva a estrutura do grupo abeliano. Isto é $[n]\,(P \oplus Q) = [n]\,P \oplus [n]\,Q$ e $[n]\,\mathcal{O} = \mathcal{O}$. De facto esta função pertence a uma classe de transformações entre pontos que se designa por **isogenias** que, por seu lado, estão dentro da classe dos **homomorfismos** de curvas.

200 Noção

Dada uma curva elíptica $C = E/K$ *uma* **aplicação racional** *é um triplo de polinómios homogéneos do mesmo grau* $\phi = [\phi_1, \phi_2, \phi_3] \in \overline{\mathbb{K}}[x, y, z]^3$ *em que pelo menos um deles não pertence a* I_C .

A aplicação ϕ *é equivalente à aplicação* $\psi = [\psi_1, \psi_2, \psi_3]$, *e escreve-se* $\phi \cong \psi$, *quando,*

$$\phi_i\,\psi_j - \phi_j\,\psi_i \;\in\; I_{E/K} \qquad \textit{para todo} \quad i, j \in 1, 2, 3$$

As classes de equivalência definidas no espaço das aplicações racionais pela relação de equivalência $\cong$, *formam a espaço dos* **homomorfismos** $\mathrm{Homo}(C)$.

É fácil verificar que, dado um qualquer ponto da curva $P \in C$, cada homomorfismos $\phi \in \mathrm{Homo}(C)$ associa P a um ponto bem definido do espaço $\mathbb{P}^2$. Para a determinação desse ponto é indiferente qual a aplicação racional que se escolhe (dentro da mesma classe) ou qual o triplo de coordenadas que usamos para P. Fica definida assim uma função $\phi\colon C \to \mathbb{P}^2$.

Interessa-nos estender esta noção da definir homomorfismos entre duas curvas elípticas. Temos assim

201 Noção

No seguimento da definição 200 seja C' *for uma segunda curva elíptica sobre o mesmo corpo* K. *Um homomorfismo* ϕ *determina um* **homomorfismo de curvas** *se se verifica* $\phi(P_\infty) = P_\infty$ *e*

$$p(\phi_1(x, y, z)\ ,\ \phi_2(x, y, z)\ ,\ \phi_3(x, y, z)) \in I_C \qquad \textit{para todo} \quad p \in I_{C'} \tag{155}$$

O conjunto dos homomorfismos entre C *e* C' *representa-se por* $\mathsf{Homo}(C, C')$.

Se $\phi \in \mathsf{Homo}(C, C')$ *é surjectivo então designa-se por* **isogenia** *entre curvas.*

As condições aqui impostas, para além de assegurar que o ponto no infinito seja sempre mapeado no ponto no infinito, asseguram que qualquer ponto $P \in C$ é mapeado num ponto $\phi(P) \in C'$.

O resultado fundamental, que se pode provar recorrendo aos divisores, é

202 TEOREMA
Se $\phi \colon C \to C'$ *é uma isogenia entre curvas elípticas* C *e* C' *então* ϕ *preserva a estrutura dos grupos abelianos.*

Vimos que as funções $[n]\cdot \colon C \to C$ são isogenias dentro da mesma curva C. Um outro exemplo particularmente importante de isogenia é a **isogenia de Frobenius**.

Recordemos que K é um corpo finito, por hipótese. Assim terá a forma $K = \mathbb{F}_{p^d}$ em que o primo p é a sua característica e d a sua sua dimensão. Recordemos que a função $\sigma_p \colon x \mapsto x^p$ é designada por **morfismo de Frobenius** no corpo K e é um endomorfismo (preserva a estrutura algébrica e é um isomorfismos) que fixa os elementos de $\mathbb{F}_p$ nesse corpo.

□

No que se segue vamos considerar uma curva elíptica $C = E/K'$ definida numa extensão K' de K. Note-se que K' continua a ter característica p e o polinómio que determina a curva é um elemento de $K[x, y, z]$.

A **isogenia de Forbenius** estende este morfismo aplicando σ_p componente-a-componente. Ou seja

$$\sigma_p : (x, y, z) \mapsto (x^p , y^p , z^p) \tag{156}$$

Não é verdade, normalmente, que σ_p seja uma isogenia de uma curva C para a mesma curva C. No entanto é fácil construir uma segunda curva C' de tal forma que σ_p seja uma isogenia entre estas duas curvas.

EXEMPLO 43: Suponhamos que $K = \mathbb{F}_{2^n}$ é um corpo de característica 2 e que que C é uma curva elíptica cuja componente afim é determinada pelo polinómio

$$y^2 + x\,y + x^3 + \upsilon \qquad \text{com} \quad \upsilon \in K$$

Se for $y^2 + x\,y + x^3 + \upsilon = 0$, elevando ao quadrado temos

$$(y^2)^2 + (x^2)\,(y^2) + (x^2)^3 + \upsilon^2 = 0$$

Então, se (x, y) define uma raiz do polinómio original, o par (x^2, y^2) define uma raiz do polinómio $y^2 + x\,y + x^3 + \upsilon^2$ que é idêntico ao anterior excepto no facto de a constante μ ser substituída por μ^2.

Porém, considerando $K = \mathbb{F}_p$ e atendendo que E/K' é definida por um polinómio $p \in \mathbb{F}_p[x, y, z]$ (i.e, todos os coeficientes do polinómio pertencem a $\mathbb{F}_p$), então, dado que σ_p fixa os elementos de $\mathbb{F}_p$, a isogenia de Frobenius

$(x, y) \mapsto (x^p, y^p)$ preserva o polinómio. Por isso σ_p é, neste caso, uma **endo-isogenia** ; isto é, uma isogenia da curva C para, de novo, a curva C.

Uma generalização simples consiste em considerar o homomorfismo σ_P^d (a potência de ordem d da isogenia de Frobenius). Neste caso qualquer polinómio $p \in K[x, y, z]$ é fixado por esta aplicação e, por isso, ela define uma endo-isogenia.

Note-se que, como as coordenadas x, y estão contidas na extensão K', elas não são fixadas pelo morfismo; por isso, normalmente, será $(x, y) \neq (x^{p^d}, y^{p^d})$.

EXEMPLO 44: Neste exemplo vamos usar curvas elípticas sobre o corpo $\mathbb{F}_4$ e sobre a sua extensão $\mathbb{F}_{16}$. Para determinar os elementos de $\mathbb{F}_4$ usaremos polinómio característico $\beta^2 + \beta + 1$. Assim os elementos de $\mathbb{F}_4$ serão $\{0, 1, \beta, 1 + \beta\}$.

Considere-se agora a curva $E/\mathbb{F}_4 : y^2 + x\,y + x^3 + 1$.

Pode-se verificar que esta curva tem 7 pontos afins, para além do ponto no infinito P_∞. São eles

$$\{ (0,1) \ , \ (1,0) \ , \ (1,1) \ , \ (\beta,0) \ , \ (\beta,\beta) \ , \ (\beta+1,0) \ , \ (\beta+1,\beta+1) \}$$

O morfismo de Frobenius é, neste caso, $x \mapsto x^2$. Notando que $\beta^2 = \beta + 1$ e que $(\beta + 1)^2 = \beta$ neste corpo, a

isogenia de Frobenius $(x, y) \mapsto (x^2, y^2)$ mapeia os pontos da lista anterior, respectivamente, em

$$\{ (0,1) \ , \ (1,0) \ , \ (1,1) \ , \ (\beta+1, 0) \ , \ (\beta+1, \beta+1) \ , \ (\beta, 0) \ , \ (\beta, \beta)\}$$

Portanto, neste caso, a isogenia mapeia pontos da curva em pontos da mesma curva.

Considere-se agora uma outra curva $E/\mathbb{F}_4 : y^2 + x\,y + x^3 + \beta$.

Os seus pontos afins são apenas

$$\{(0, \beta+1) \ , \ (\beta, 1) \ , \ (\beta, \beta+1)\}$$

A isogenia de Frobenius $(x, y) \mapsto (x^2, y^2)$ estabelece-se entre esta curva e a curva $E/\mathbb{F}_4 : y^2 + x\,y + x^3 + (\beta+1)$ cujos pontos são

$$\{(0, \beta) \ , \ (\beta+1, 1) \ , \ (\beta+1, \beta)\}$$

□

Vamos agora considerar curvas sobre a extensão $\mathbb{F}_{16}/\mathbb{F}_4$ usando o polinómio característico $\alpha^4 + \alpha + 1$.

A imersão de $\mathbb{F}_4$ em $\mathbb{F}_{16}$ faz-se pelo morfismo que mapeia $\beta \mapsto \alpha^2 + \alpha$; de facto, calculando $\beta^2 + \beta + 1$ tem-se

$$\beta^2 + \beta + 1 \quad \rightarrow \quad (\alpha^4 + \alpha^2) + (\alpha^2 + \alpha) + 1 \quad \rightarrow \quad \alpha^4 + \alpha + 1 \quad \rightarrow \quad 0$$

Vamos considerar, nesta extensão, curvas definidas pelos dois polinómios considerados atrás (tendo em atenção a substituição $\beta \longrightarrow \alpha^2+\alpha$). É óbvio que as curvas $E/\mathbb{F}_{16} : y^2+x\,y+x^3+1$ e $E/\mathbb{F}_{16} : y^2+x\,y+x^3+(\alpha^2+\alpha)$ contêm todos os pontos das curvas correspondentes em $\mathbb{F}_4$ e ainda alguns pontos adicionais.

A componente afim de $E/\mathbb{F}_{16} : y^2 + x\,y + x^3 + 1$ contém 15 pontos em vez dos 7 pontos em $\mathbb{F}_4$

$(0,1)$, $(1,0)$, $(1,1)$, $(\alpha^3, \alpha^2+\alpha+1)$, $(\alpha^3, \alpha^3+\alpha^2+\alpha+1)$, $(\alpha^2+\alpha, 0)$, $(\alpha^2+\alpha, \alpha^2+\alpha)$, $(\alpha^3+\alpha^2, \alpha^2+\alpha)$, etc

Na componente afim de $E/\mathbb{F}_{16} : y^2 + x\,y + x^3 + \alpha^2 + \alpha$ a diferença é mais substancial; em vez dos 3 pontos em $\mathbb{F}_4$ tem-se 23 pontos em $\mathbb{F}_{16}$.

Ambos os polinómios mantêm-se invariantes pela transformação $x \mapsto x^4$; por isso o morfismo $(x, y) \mapsto (x^4, y^4)$ é uma endo-isogenia.

Regressemos às isogenias sobre uma curva elíptica $E/K : \phi$, com $\phi \in K_0[x, y, z]]$ e K um extensão K/K_0 . Seja p a característica de K_0 (e também de K).

203 NOÇÃO
O núcleo de $[n]\colon E/K \longrightarrow E/K$ *(isto é, o conjunto* $\{\, P \in E/K \mid [n]P = \mathcal{O} \,\}$*) é representado por* $E/K[n]$ *e designa-se por* **grupo de torção de** E/K **de ordem** n*. Os pontos de* $E/K[n]$ *designam-se por* K-**pontos de torção** *de ordem* n*.*

O seguinte resultado é fundamental ao nosso estudo.

204 **Teorema**

Se K é algebricamente fechado e a característica p é primo relativamente a n, então $E/K[n] \simeq \mathbb{Z}_n \times \mathbb{Z}_n$. Se $n = p^s$, para algum $s \geq 1$, então um de dois casos ocorre: ou $E/K[n] = \{\mathcal{O}\}$ ou então $E/K[n] \simeq \mathbb{Z}_{p^s}$.

Se ocorrer a primeira hipótese $E/K[p^s] = \{\mathcal{O}\}$, a curva diz-se **super-singular** e esta propriedade é independente do valor de s. No caso contrário ($E/K[p^s] \simeq \mathbb{Z}_{p^s}$) a curva diz-se **ordinária**.

É importante notar-se que este teorema exige que K coincida com $\bar{K_0}$ (o fecho algébrico do corpo onde característica p onde o polinómio gerador ϕ está definido). Quando K é uma outra qualquer extensão de K_0, contida em $\bar{K_0}$, o K-grupo de torção tem, naturalmente, menos pontos.

Obviamente, se escolhermos um qualquer $P \neq \mathcal{O} \in E/K[n]$, qualquer ponto $Q = [k]P$ continua a ser um elemento $E/K[n]$. Por isso a órbita de P é um sub-grupo cíclico de $E/K[n]$.

Sabe-se, da teoria genérica dos grupos de torção, que $E/K[n]$ é a soma directa dos seus subgrupos primos. Isto é a soma directa de subgrupos de ordem da forma q^s, sendo q um primo. Nomeadamente em criptografia são extremamente importantes os sub-grupos de ordem prima dos grupos de torção.

8.5 Aritmética nas Curvas Elípticas

Vimos na secção anterior que uma **curva elíptica** sobre o corpo $\mathbb{K}$ é determinada, em coordenadas afins, por um polinómio $\phi \in \overline{\mathbb{K}}[x, y]$ da forma de Weitstraß que, de forma simplificada, se pode escrever

$$\phi = y^2 + y\,h(x) + f(x) \tag{157}$$

com $h, g \in \mathbb{K}[x]$ sendo $\deg(h) \leq 1$ e $\deg(f) = 3$ de tal forma que as derivadas $\partial\phi/\partial x$ e $\partial\phi/\partial y$ não se anulam simultâneamente.

Vimos também que as raízes deste polinómio definem a chamada **parte afim** da curva. Existe ainda um outro ponto da curva, que representamos por P_∞ ou $\mathcal{O}$, que completa a curva e que não pertence à parte afim.

□

Sendo $h = a_1\,x + a_3$ e $f = x^3 + a_2\,x^2 + a_4\,x + a_6$, a condição de não anulação simultânea das derivadas parciais ocorre se e só se

$$16\,{a_2}^2 - 8\,a_2{a_1}^2 + {a_1}^4 - 48\,a_4 + 24\,a_1a_3 \;<\; 0$$

Se $\mathbb{K}$ tem característica diferente de 2 ou 3 então, através de substituição de variáveis $y \leftarrow 2\,y + h$ e $f \leftarrow f - h^2/4$, transforma a forma genérica de Weitsrass em (157) simplificando-a em

$$\phi = y^2 + f(x) \tag{158}$$

Se $\mathbb{K}$ tem característica 3 a mudança de variáveis $y \leftarrow y + h$ e $f \leftarrow f + h^2$ conduz à mesma forma equivalente $\phi = y^2 + f(x)$. Em ambos os casos ϕ define uma curva elíptica se e só se f não tem raízes múltiplas; isto é, todas as raízes de f têm de ser distintas.

Quando a característica do corpo K é 2, a forma de Weierstraß pode ainda ser simplificada. Para isso vamos definir $\xi \doteq \partial h/\partial x$. Como h tem grau 0 ou 1, o valor ξ é um elemento de $\mathbb{K}$ que pode ser nulo (se h tiver grau 0) ou não. Vamos considerar separadamente estes dois casos.

$\xi = \partial h/\partial x = 0$ Estas curvas dizem-se **super-singulares**.

Neste caso $h(x)$ é uma constante e, escrevendo $f(x) = x^3 + a\,x^2 + b\,x + c$, pode-se efectuar a mudança de variáveis $x \leftarrow x + a$, $\mu = a^2 + b$, $\upsilon = ab + c$ que conduz à curva equivalente

$$\phi = y^2 + hy + x^3 + \mu x + \upsilon \tag{159}$$

Nesta curva tem-se $\partial\phi/\partial y = h$ e a curva será não-singular (isto é, ϕ determina uma curva elíptica) se e só se for $h \neq 0$.

$\xi = \partial h/\partial x \neq 0$ Neste caso tem-se $h(x) = \kappa + \xi\,x$. A curva diz-se **ordinária**.

Usando a mesma representação de f existe uma mudança de variáveis que conduz à forma simplificada,

$$\phi = y^2 + xy + x^3 + \mu x^2 + \upsilon \tag{160}$$

Aqui $\partial\phi/\partial x = y + x^2$ e $\partial\phi/\partial y = x$. Isto significa que a curva é não-singular desde que não contenha o ponto $(0,0)$.

Para as curvas ordinárias existe ainda uma simplificação adicional. Suponhamos que se toma $c = \mu$ ou o seu complemento $c = 1 + \mu$ consoante o traço de μ é 0 ou 1. Em qualquer dos casos tem-se $\mathrm{Tr}(c) = 0$.

Neste caso a equação $\lambda^2 + \lambda + c = 0$ tem solução. A mudança de variáveis $y \mapsto y + \lambda\, x$ conduz à curva equivalente

$$\phi' \;=\; y^2 + xy + x^3 + (c+\mu)\, x^2 + \upsilon$$

Escolhendo $c = \mu$ ou $c = 1 + \mu$ obtém-se as curvas

$$\begin{aligned} \phi &= y^2 + xy + x^3 + \upsilon \qquad \text{ou} \\ \phi &= y^2 + xy + x^3 + x^2 + \upsilon \end{aligned} \tag{161}$$

que são as formas mais genéricas de curvas ordinárias em corpos de característica 2. Qualquer destas formas pode ser adoptada escolhendo uma ou outra consoante se pretende um coeficiente de x^2 com traço 0 ou com traço 1.

□

Na secção anterior vimos que as curvas elípticas $E/\mathbb{K}$ se identificam, no contexto geral do estudas das curvas

planas, com as curvas de genus 1 que têm um ponto identificado $\mathcal{O}$. Desta interpretação resultaram, na secção anterior, dois resultados essenciais sobre a estrutura algébrica das curvas elípticas:

1. Em primeiro lugar a proposição 186 (ver pag. 495) identifica em cada curva $E/\mathbb{K}$ a estrutura de um grupo abeliano $\langle E/\mathbb{K}\ ,\ \oplus\ ,\ \mathcal{O}\rangle$.
2. Em segundo lugar que a operação de grupo $\oplus$ tem uma simples interpretação geométrica; o lema 196 diz-nos que, se for $S = P \oplus Q$ então $P, Q, -S$ estão sobre uma mesma recta (são colineares). O mesmo lema diz-nos também que, para todo ponto P, o triplo de pontos $P, -P, \mathcal{O}$ também está sempre sobre uma mesma recta. Se P tiver coordenadas $[P_1, P_2, P_3]$, tem-se $P \otimes \mathcal{O} = [P_3, 0, -P_1]$ e a recta respectiva será $xP_3 - zP_1$

Esta interpretação geométrica permite facilmente encontrar fórmulas explícitas para calcular as coordenas afins de $P \oplus Q$ e de $-P$ em função das coordenadas afins de P e Q. Assumimos a forma genérica de Weierstraß em coordenadas afins

$$y^2 + a_1\, x\, y + a_3\, y - x^3 - a_2\, x^2 - a_4\, x - a_6 \tag{162}$$

1. **Determinar $-P$**
   Se P tem coordenadas afins $[p_1, p_2]$ então a recta $\mathbf{l}(P \otimes \mathcal{O})$ é $xp_3 - zp_1$.
   O terceiro ponto de intersecção com a curva é $-P$. Atendendo a que P pertence à curva, verifica-se que

$$-P \ = \ (p_1, -\mu) \qquad \text{sendo} \quad \mu = p_2 + a_1\, p_1 + a_3 \tag{163}$$

2. **Determinar $P \oplus Q$**
   Sekam $P = (p_1, p_2)$ e $Q = (q_1, q_2)$ as coordenadas afins dos pontos.
   Determina-se a recta $p = \mathbf{l}(P \oplus Q)$ que passa por P e Q. Caso seja $P = Q$ essa recta é a tangente à curva em P. Determina-se o terceiro ponto R de intersecção da recta com a curva e faz-se $P \oplus Q = -R$.
   Temos duas situações para cálculo do declive λ da recta p.

$$\lambda = (p_2 - q_2)/(p_1 - q_1) \qquad \text{quando} \quad P \neq \pm Q \tag{164}$$

$$\lambda = (3\,p_1^2 + 2\,a_2\,p_1 + a_4 - p_2)/(2\,p_2 + a_1\,p_1 + a_3) \qquad \text{quando} \quad P = Q \tag{165}$$

   Então

$$P \oplus Q = (\,\eta\ ,\ \lambda\,(p_1 - \eta) - \mu\,) \tag{166}$$

$$\text{sendo} \quad \eta = \lambda^2 + a_1\,\lambda - a_2 - p_1 - p_2 \qquad \text{e} \qquad \mu = p_2 + a_1\,\eta + a_3$$

Esta é a forma mais geral para determinar o valor da soma $P \oplus Q$. Consoante a carcaterística do corpo K e as formas simplificadas de curvas que daí resultam várias simplificações e optimizações são possíveis.

8.6 Emparelhamentos de Tate e Weil

Sejam G_1, G_2 dois sub-grupos de curvas elípticas ambos de expoente n; isto é, para todo $p \in G_1, G_2$ tem-se $[n]P = \mathcal{O}$. Considere-se ainda um grupo cíclico Γ de ordem n escrito de forma multiplicativa. Recordemos que

205 NOÇÃO

Um **emparelhamento** *é uma função* $e\colon G_1 \times G_2 \to \Gamma$ *que é*

- ***bilinear:*** *para todo* $P, P' \in G_1$ *e* $Q, Q' \in G_2$ *tem-se* $e(P \oplus P', Q) = e(P,Q) \cdot e(P',Q)$ *e* $e(P, Q \oplus Q') = e(P,Q) \cdot e(P,Q')$.
- ***não-degenerado:*** *para todo* $P \neq \mathcal{O} \in G_1$ *existe* $Q \in G_2$ *tal que* $e(P,Q) \neq 1$ *e, dualmente, para todo* $Q \neq 0 \in G_2$ *existe* $P \in G_1$ *tal que* $e(P,Q) \neq 0$.

Como consequência imediata da definição, temos

206 FACTO

Se $e\colon G_1 \times G_2 \to \Gamma$ *é um emparelhamento, então para todo* $P \in G_1$ *e todo* $Q \in G_2$

- $e(P,0) = e(0,Q) = 1$.
- $e(-P,Q) = e(P,-Q) = e(P,Q)^{-1}$
- $e([k]P, [j]Q = e(P\,Q)^{k\,j}$ *para todos* $k, j \in \mathbb{Z}$.

Emparelhamento de Tate

Escolha do corpo

1. *Escolha-se um primo p e um* **corpo base** $K_0 = \mathbb{F}_q$ *de característica p.*
2. *Escolha-se um primo r distinto de p. Seja k a ordem de q em $\mathbb{Z}_r^*$: isto é, o menor inteiro k tal que $q^k = 1 \mod r$. Seja $l = s \cdot (s^{-1} \mod r)$ sendo $s = (q^k - 1)/r$.*
3. *Seja $\boldsymbol{\mu}_r$ o conjunto de todas as raízes do polinómio $(X^n - 1) \in \overline{K_0}[X]$; isto é, as* **r-raízes da unidade** *de K_0.*
4. *Define-se $K = K_0(\boldsymbol{\mu}_r)$ como a menor extensão de K_0 que contém $\boldsymbol{\mu}_r$; isto é, a* **r-extensão ciclotómica** *de K_0.*
5. *Seja $(K^*)^r = \{ u^r \mid u \in K \}$ o subgrupo de K^* formado pelas r-potências de K. O grupo quociente $K^*/(K^*)^r$ é representado por K_r^*.*

O estudo das raízes da unidade, extensões ciclotómicas, polinómios ciclotómicos, etc, é uma tema da Álgebra bastante analisado[59]. Os principais resultados que nos interessam podem ser sumariados no seguinte teorema.

207 Teorema (Extensão ciclotómica)
Nas condições acima enumeradas,

[59] Consultar, por exemplo, o livro de *Steven Roman*, Field Theory, 2nd edn, Springer Verlag, nº 158 da série *Graduate Texts in Mathematics*, principalmente os Capítulos 11 e 12, para uma síntese desses resultados.

1. *O corpo K é uma extensão de grau k de $\mathbb{F}_q$ e identifica-se com $\mathbb{F}_{q^k}$.*
2. *O grupo de Galois, $\mathbb{G}(K/K_0)$ é um grupo cíclico de ordem k gerado pelo automorfismo $\sigma_q : x \mapsto x^q$.*
3. *O conjunto das r-raízes da unidade $\boldsymbol{\mu}_r$ forma um subgrupo ciclico de ordem r de K^* e o morfismo $x \mapsto x^l$ induz um isomorfismo entre o grupo quociente K_r^* e $\boldsymbol{\mu}_r$.*

Nota A relação de equivalência que gera o grupo quociente $K^*/(K^*)^r$ é, neste caso,

$$x \simeq_r y \quad \Leftrightarrow \quad (\exists u \in K^*)\,[\, x \cdot y^{-1} = u^r \,] \tag{167}$$

Pela definição de k, $(q^k - 1)$ é divisível por r; seja $s = (q^k - 1)/r$. Então, para qualquer $x \in K^*$, x^s é uma r-raíz da unidade. Sendo $l = s \cdot (s^{-1} \mod r)$, também x^l é uma r-raiz da unidade porque l é um múltiplo de s. Por outro lado, $l = 1 \mod r$ o que significa que $l - 1$ é um múltiplo de r e, por isso, $x^l \cdot x^{-1}$ é uma r-potência de um elemento de K^*. Ou seja, x e x^l são equivalentes.

Grau de embebimento

Vimos que, sendo k a ordem de q em $\mathbb{Z}_r^*$, o corpo K é uma extensão de grau k de $\mathbb{F}_q$. A constante k chama-se **grau de embebimento** de K_0 em K.

Na escolha do corpo, as várias constantes (q, r, k, l) podem-se determinar de vários modos. Por exemplo, pode-se fixar q e r e calcular grau de embebimento k como o menor inteiro k tal que r divide $q^k - 1$. Em alternativa pode-se fixar q e um grau de embebimento aceitável k, e determinar o maior primo r que divide $q^k - 1$.

Para uma implementação eficiente de emparelhamentos interessa-nos ter o grau de embebimento k tão pequeno quanto possível. Isto porque k pequeno implica que o número de elementos q^k do corpo K é relativamente pequeno e, consequentemente, são necessários poucos de bits para representar pontos de curvas elípticas E/K. Os valores ideais para k serão 2 ou 3; de facto 6 é considerado o limite superior para uma implementação razoável de emparelhamentos.

Por outro lado, r vai determinar a ordem dos grupos cíclicos e, por isso, convém que seja um primo tão grande quanto possível. Quanto maior for r mais complexo será a resolução dos problemas básicos dos grupos cíclicos (DLP, CDHP, etc.) e, por isso, mais seguras serão as técnicas criptográficas assentes nesses grupos. No mínimo r deve ser um primo só representável com 160 bits ou mais.

Estes objectivos contraditórios conduzem a uma escolha criteriosa de r que seja suficientemente grande para os grupos cíclicos serem seguros e, simultaneamente, determine uma valor de k pequeno.

EXEMPLO 45: Considere-se o corpo $K_0 = \mathbb{F}_7$. Pode-se verificar que a curva $E/\mathbb{F}_7 : y^2 - x^3 - x + 3$ tem exactamente 9 pontos e que esses pontos são gerados como múltiplos do ponto $P = (0, 2)$; os pontos $[k]P$, com $k = 1..9$, são

$$(0,2)\ ,\ (4,4)\ ,\ (5,6)\ ,\ (6,3)\ ,\ (2,0)\ ,\ (6,4)\ ,\ (5,1)\ ,\ (4,3)\ ,\ (0,5)\ ,\ P_\infty$$

Escolha-se uma ordem $r \neq p$ que seja um número primo; por exemplo $r = 13$. Neste caso $\boldsymbol{\mu}_{13}$ é formado pela unidade 1 e pelas raízes $\zeta \neq 1$ do polinómio ciclotómico $\Phi_{13}[X] = \frac{(X^{13}-1)}{(X-1)} \in \mathbb{F}_7[X]$; isto é, os ζ que verificam $\zeta^{12} + \zeta^{11} + \cdots + \zeta + 1 = 0$.

O grau de embebimento é o menor k tal que 13 divide $7^k - 1$; pode-se verificar que $k = 12$ e, portanto, K tem 7^{12} elementos. Neste caso, tem-se $l = 3194143200$. Podia-se também verificar que, aqui, tem de ser $k = r - 1$ porque o polinómio ciclotómico Φ_{13} é absolutamente irredutível em $\mathbb{F}_7$. A 13-extensão ciclotómica de $\mathbb{F}_7$, $\mathbb{F}_7(\boldsymbol{\mu}_{13})$ tem 7^{12} elementos cuja forma genérica é $a_0 + a_1\,\zeta + a_2\,\zeta^2 + \cdots + a_{11}\,\zeta^{11}$, com $a_k \in \mathbb{F}_7$.

Este valor para o grau de embebimento é péssimo: cada ponto da curva E/K exige 12 vezes mais bits que um ponto sobre a curva $E/\mathbb{F}_7$. E isto para ter apenas 13 elementos no grupo cíclico.

No entanto um grau de embebimento $k = 5$ (dentro da gama dos valores aceitáveis e bastante inferior ao valor de 12 atrás encontrado) conduz a um primo $r = 2801$ que divide $7^5 - 1$. Note-se que, mesmo para um k menor, o primo r resultante é bastante maior do que 13. O parâmetro l é também consideravelmente menor; tem-se $l = 2802 = r + 1$.

Escolha da curva

Seja $C = E/K : \phi$ uma curva elíptica cuja ordem $|E/K|$ é um múltiplo de r.

Para garantir que a ordem r divida a ordem da curva $E/K : \phi$, seria possível começar por exigir, na escolha de r, que a ordem da curva base $E/K_0 : \phi$ fosse já um múltiplo de r. Porém isso implica que o corpo de base K_0 tivesse, já à partida, um número de elementos suficientes para que tal ocorra.

Em alternativa pode-se tentar escolher (eventualmente, por tentativas sucessivas) a ordem r, o grau de embebimento k e o polinómio ϕ de modo que todas estas condições ocorram: o grau de embebimento k é pequeno, o corpo base $\mathbb{F}_q$ é pequeno, r é grande e r divide a ordem da curva definida no corpo $\mathbb{F}_{q^k}$.

□

A ordem r determina em C vários subgrupos com interesse:

- O grupo de torção $C[r] = \{ P \in C \mid [r]P = \mathcal{O} \}$.
- O grupo $r\,C = \{[r]U \mid U \in C\}$ dos r-múltiplos de pontos da curva; $r\,C$ é a imagem da isogenia $[r]\cdot$.
- O grupo quociente $C_r = C/rC$.

Note-se a analogia entre C_r e o grupo K_r^*: os elementos de C_r são as classes de equivalência geradas em C, pela relação $P \cong Q \Leftrightarrow (\exists U \in C)\,[\,P - Q = [r]U\,]$. Note-se também que as condições impostas em r e p, implicam que todos os grupos $C[r], C_r, K_r^*$ e $\boldsymbol{\mu}_r$ têm exactamente r elementos.

208 DEFINIÇÃO (EMPARELHAMENTO DE TATE)
O ***emparelhamento de Tate*** *e ordem r em C, é a função*

$$\mathbf{e}_r : C[r] \times C_r \longrightarrow \boldsymbol{\mu}_r$$

tal que, para todo ponto $P \in C[r]$ e classe de equivalência $q \in C_r$, o valor de $\mathbf{e}_r(P, q)$ é o elemento de $\boldsymbol{\mu}_r$ gerado da seguinte forma:

1. *Determina-se $f \in K(C)$ tal que $(f) = r(P \oplus R) - r(R)$, para algum ponto $R \in C$.*

Porque $P \in C[r]$, a função (f) existe. Note-se que P verifica $[r]P = \mathcal{O}$ e, pela definição 185, tem-se $r(P) - r(\mathcal{O}) \sim 0$; logo, para todo R, será $r(P \oplus R) - r(R) \sim 0$. Usualmente escolhe-se $R = \mathcal{O}$.

2. *Escolhe-se $Q \in q$ e determina-se um divisor $D \sim (Q) - (\mathcal{O})$ que tenha um suporte disjunto do de (f).*

   Normalmente, basta escolher um ponto $S \in C$, tal que tanto S como $Q \oplus S$ não sejam nem zero nem pólo de f, e definir $D = (Q \oplus S) - (S)$. Pela definição 185, tem-se $(Q \oplus S) - (\mathcal{O}) \sim (Q) + (S) - 2\,(\mathcal{O})$ e, portanto, $(Q \oplus S) - (S) \sim (Q) - (\mathcal{O})$.

3. *Determina-se $f(D)$ e define-se $\mathbf{e}_r(P, q) = f(D)^l$; ou seja, como a r-raiz da unidade equivalente a $f(D)$.*

   Como D não contém pólos ou zeros de f, a função f é regular em D e tem-se $f(D) \neq 0$; logo $f(D) \in K^*$. O valor $f(D)^l$ determina a raiz da unidade $\mu \in \boldsymbol{\mu}_r$ tal que $f(D) \simeq_r \mu$.

No passo (1) temos várias funções racionais $f \in K(C)$ que verificam $(f) = r(P \oplus R) - r(R)$. O valor de $\mathbf{e}(P, q)$ deve ser independente do ponto R e da função f escolhidas. É também importante ter em atenção, no passo (2), que Q é um representante da classe de equivalência $q \in C_r$ e o valor $f(D)$ depende de Q; são possíveis vários pontos Q dentro da mesma classe de equivalência $q \in C_r$ e vários divisores $D \sim (Q) - (\mathcal{O})$. Escolhendo um diferente $Q' = Q \oplus [r]U$ e um diferente $D' \sim (Q') - (\mathcal{O})$, o valor para $f(D')$ e o valor de $f(D)$ devem conduzir, no final, ao mesmo valor de $\mathbf{e}_r(P, q)$.

Em resumo: para a correcção do resultado de $\mathbf{e}(P,q)$, deve ser indiferente o ponto R, a função f, o representante Q e o divisor D, escolhidos desde que verifiquem as condições estipuladas na respectiva escolha. No entanto, a liberdade na selecção de determinados valores específicos de R, f, Q e D é importante para a eficiência da implementação deste algoritmo.

Todos estes factos levam a que o resultado $f(D)$ seja visto como um representante de uma classe de equivalência em K_r^* e, consequentemente, um elemento de $\boldsymbol{\mu}_r$. A razão porque o algoritmo na noção 208 define correctamente uma função $C[r] \times C_r \longrightarrow \boldsymbol{\mu}_r$ é resultado dos seguintes lemas.

209 LEMA *Sejam $f, g \in K(C)$ funções racionais homogéneas tais que $(f) = r(P) - r(\mathcal{O})$ e, para um qualquer ponto $R \in C$, $(g) = r(P \oplus R) - r(R)$. Então, para todo o divisor de grau zero D sobre K com suporte disjunto do de (f) e do de (g), tem-se $f(D) \simeq_r g(D)$.*

210 LEMA *Seja $f \in K(C)$ tal que que $(f) = r(P) - r(\mathcal{O})$. Sejam D e D' divisores sobre K de grau zero tais que $D \sim D'$, e com suporte disjunto do de (f). Então $f(D) \simeq_r f(D')$.*

211 LEMA *Seja $f \in K(C)$ funções racional homogénea tal que $(f) = r(P) - r(\mathcal{O})$. Sejam D e D' divisores sobre K de grau zero tais que $D \sim (Q) - (\mathcal{O})$ e $D' \sim (Q \oplus [r]U) - (\mathcal{O})$, cujo suporte é disjunto do de (f). Então $f(D) \simeq_r f(D')$.*

Prova As provas destes três lemas são uma simples aplicação dos resultados da teoria dos divisores sobre curvas elípticas. Note-se que, em todas as provas, as funções racionais f e g estão sempre definidas, na curva, a menos da multiplicação por uma constante $\lambda \neq 0$. No entanto, pelo facto 189, quando aplicadas a divisores D de grau zero, os valores de $f(D)$ e $g(D)$ são independentes dessa constante.

Lema 209 Pela definição 185 temos $(P \oplus R) - (P) - (R) + (\mathcal{O}) \sim 0$. Seja $u \in K(C)$ tal que $(u) = (P \oplus R) - (P) - (R) + (\mathcal{O})$; então $(u^r) = r(u) = r(P \oplus R) - r(R) - r(P) + r(\mathcal{O})$ e consequentemente $(g) = (u^r) + (f) = (u^r\, f)$. Donde, para alguma constante $\lambda \neq 0$, tem-se $\lambda\, g(X) = u(X)^r\, f(X)$, para todo $X \in C$ onde g e f sejam regulares. Para qualquer divisor cujo suporte seja disjunto do suporte de (f) e de (g) temos $g(D) = f(D)\, u(D)^r$ e, consequentemente, $g(D) \simeq_r f(D)$.

Lema 210 Seja $u \in K(C)$ tal que $D = D' + (u)$. Então o suporte de u é disjunto do de f e $f(D) = f(D')\, f((u))$. Pela reciprocidade de Weil (teorema 190) temos $f((u)) = u((f)) = u(r(P) - r(\mathcal{O})) = u(P)^r / u(\mathcal{O})^r$. Consequentemente $f(D) \simeq_r f(D')$.

Lema 211 Temos $D' - D \sim (Q \oplus [r]U) - (Q) \sim r(U) - r(\mathcal{O})$ e, portanto, como o suporte de (f) é disjunto do de D e D', temos $f(D')/f(D) = f(U)^r / f(\mathcal{O})^r$. Logo $f(D') \simeq_r f(D)$.

Estamos agora em condições de provar o resultado fundamental.

212 Teorema (Emparelhamento de Tate)
O algoritmo apresentada na definição 208 determina uma função $\mathbf{e}_r : C[r] \times C_r \longrightarrow \boldsymbol{\mu}_r$ *e esta função:*

1. *É um emparelhamento (noção 205),*
2. *Para todo o automorfismo* $\sigma \in \mathbb{G}(K/K_0)$, *verifica* $\mathbf{e}_r(\sigma(P), \sigma(q)) = \sigma(\mathbf{e}_r(P, Q))$.

Esboço de Prova Os lemas (209), (210) e (211), dizem-nos que, independentemente do ponto R, da função f, do representante $Q \in q$ e do divisor D, $f(D)$ é sempre um elemento da mesma classe de equivalência em K_r^*; o isomorfismo entre K_r^* e $\boldsymbol{\mu}_r$ completa a definição da função.

Para provar (1) (isto é, ver que $(P,q) \mapsto \mathbf{e}_r(P,q)$ é um emparelhamento de acordo com a noção 205) temos de provar que é bilinear e não-degenerada.

Biliear no 1^{o} argumento, $\mathbf{e}(P \oplus P', q) = \mathbf{e}(P,q) \cdot \mathbf{e}(P',q)$: Sejam $f, f' \in K(C)$ tais que $(f) = r(P) - r(\mathcal{O})$ e $(f') = r(P') - r(\mathcal{O})$. Seja $u \in K(C)$ tal que $(u) = (P \oplus P') - (P) - (P') + (\mathcal{O})$ e seja $h = f \cdot f' \cdot u^r$. Então $(h) = (f) + (f') + r(u) = r(P \oplus P') - r(\mathcal{O})$. Para um qualquer divisor $D \sim (Q) - (\mathcal{O})$, com $Q \in q$, com suporte disjunto do de h, tem-se $h(D) \simeq_r f(D)\, f'(D)$ e este valor determina $\mathbf{e}_r(P \oplus P', q)$; como $f(D)$ e $f'(D)$ determinam, respectivamente, $\mathbf{e}_r(P,q)$ e $\mathbf{e}_r(P,q)$, tem-se $\mathbf{e}_r(P \oplus P', q) = \mathbf{e}_r(P,q) \cdot \mathbf{e}_r(P',q)$.

As restantes 3 condições do emparelhamento (bilinear no 2^{o} argumento e não-degenerado no 1^{o} e 2^{o} argumentos) provam-se de forma semelhante.

Para provar (2) (invariância por automorfismos de Galois), basta ver que $\sigma(f(P)) = f_\sigma(\sigma(P))$, sendo f_σ a aplicação do automorfismo σ aos coeficientes de f. Se $(f) = r(P) - r(\mathcal{O})$ então $(f_\sigma) = r(\sigma(P)) - r(\mathcal{O})$. Sendo $D \sim (Q) - (\mathcal{O})$, tem-se $\mathbf{e}_r(\sigma(P), \sigma(Q)) = f_\sigma(\sigma(D)) = \sigma(f(D)) = \sigma(\mathbf{e}_r(P,Q))$.

Emparelhamento de Weil

Fixemos um corpo base $K_0 = \mathbb{F}_q$, e uma ordem $r \neq p$. Como anteriormente $\boldsymbol{\mu}_r$ é o grupo cíclico das r-raízes da unidade em K_0.

Seja $E/\overline{K_0}$ uma curva elíptica sobre o fecho algébrico de K_0, cuja ordem $|E/\overline{K_0}|$ é um múltiplo de r. Como habitualmente $E/\overline{K_0}[r]$ denota o grupo de torção de ordem r.

Escolhe-se K como a mínima extensão de K_0 que contém as coordenadas de todos os pontos neste grupo de torção. Abusando um pouco da notação podemos escrever $K = K_0(E/\overline{K_0}[r])$.

213 NOÇÃO (EMPARELHAMENTO DE WEIL)
O ***emparelhamento de Weil*** *é a função* $\mathbf{w}_r \colon E[r] \times E[r] \to \boldsymbol{\mu}_r$ *com* $\mathbf{w}_r(P,Q)$ *gerado por:*

1. *Escolhe-se dois divisores* $D \sim (P) - (\mathcal{O})$ *e* $D' \sim (Q) - (\mathcal{O})$ *de suporte disjuntos.*

   Basta escolher pontos R e S apropriados e fazer $D = (P \oplus R) - (R)$ e $D' = (Q \oplus S) - (S)$.

2. *Escolhe-se funções* $f, g \in K(E)$ *tais que* $(f) = r\,D$ *e* $(g) = r\,D'$.

   Porque P e Q pertencem a $E[r]$ tem-se $[r]P = [r]Q = \mathcal{O}$. Logo $r(P) - r(\mathcal{O}) = r\,D \sim 0$ e $r(Q) - r(\mathcal{O}) = r\,D' \sim 0$. Consequentemente, existem $f, g \in K(E)$ tais que $(f) = r\,D$ e $(g) = r\,D'$.

3. *Calcula-se* $w = f(D')/g(D)$ *e faz-se* $\mathbf{w}_r(P,Q) = w^l$.

   Pelo lema 210, tem-se $f(D' + (h')) \simeq_r f(D')$ e $g(D + (h)) \simeq_r g(D)$ para quaisquer $h, h' \in K(E)$. Portanto, o valor de $\mathbf{w}_r(P,Q)$ é independente dos divisores D, D' desde que satisfaçam as condições em (1).

Como sabemos, funções racionais distintas com o mesmo divisor são determindas a menos da multiplicação por uma constante; isso não afecta o resultado da sua aplicação a divisores de grau 0. Por isso o valor de $\mathbf{w}_r(P,Q)$ é independente da escolha específica de funções f, g em (2).

As propriedades da função $\mathbf{w}_r$ podem ser sumariadas no seguinte resultado.

214 TEOREMA (EMPARELHAMENTO DE WEIL)
O algoritmo apresentado na noção 213 define um emparelhamento. Adicionalmente verifica:

1. *(*Invariância de Galois*) Para todo automorfismo* $\sigma \in \mathbb{G}(\overline{K}/K)$ *verifica-se*

$$\mathbf{w}_r(\sigma(P), \sigma(Q)) \;=\; \sigma(\mathbf{w}_r(P,Q))$$

2. *(*Simetria*)* $\mathbf{w}_r(,P,P) = 1$ *e* $\mathbf{w}_r(,P,Q) = \mathbf{w}_r(,Q,P)^{-1}$.

Esboço de prova A prova de que a função é um emparelhamento usa as mesmas técnicas que a prova equivalente no emparelhamento de Tate. A invariância de Galois é também provada do mesmo mdodo. A simetria é óbvia a partir da definição.

Existem semelhanças óbvias entre os emparelhamentos de Tate e de Weil. Para pontos P, Q na intersecção dos respectivos domínios é óbvio, pela definição, que

$$\mathbf{w}_r(P,Q) \quad = \quad \frac{\mathbf{e}_r(P,Q)}{\mathbf{e}_r(Q,P)} \tag{168}$$

Existem, no entanto, alguns detalhes que podem impedir esta relação óbvia. Em primeiro lugar, as curvas E/K são definidas sobre corpos K distintos. Depois os próprios domínios das funções são subgrupos distintos da curva.

Em ambos os casos o corpo de base K_0 é $\mathbb{F}_q$ com característica p; em ambos os casos a ordem r é um primo distinto de p. Porém,

(i) no emparelhamento de Tate, escolhe-se primeiro o corpo K como $\mathbb{F}_q(\boldsymbol{\mu}_r)$ e escolhe-se depois a curva E/K de forma a ter uma ordem que seja múltipla de r.

(ii) no emparelhamento de Weil, escolhe-se primeiro uma curva $E/\overline{\mathbb{F}_q}$ que tenha uma ordem múltipla de r e fixa-se o corpo K como a extensão $\mathbb{F}_q(E[r])$.

Geralmente, o corpo $\mathbb{F}_q(E[r])$ seria muito maior do que o corpo $\mathbb{F}_q(\boldsymbol{\mu}_r)$. No entanto, para muitas situações com interesse criptográfico, os dois coincidem. Isto é resultado do seguinte teorema.

215 Teorema (Balasubramanian e Kobliz)
Seja $E/\mathbb{F}_q$ uma curva elíptica definida dobre um corpo $\mathbb{F}_q$ de característica p. Seja $r \neq p$ um primo que não divide $q-1$ mas divide a ordem $|E/\mathbb{F}_q|$ da curva. Então $E[r] \subset E/\mathbb{F}_{q^k}$ se e só se r divide $q^k - 1$.

Note-se que as condições do teorema exigem que, à partida, a curva sobre o corpo base já contenha pontos suficientes para conter $E[r]$; nomeadamente a sua ordem tem de ser um múltiplo de r. Esta condição é mais forte das que foram colocadas na definição de curva em ambos os emparelhamentos, onde se requeria que apenas a curva sobre a extensão k verificasse uma condição análoga.

Se estas condições forem verificadas o domínio de ambos emparelhamentos é compatível (tendo em atenção que, no emparelhamento de Tate, os dois argumentos têm domínios diferentes) e a igualdade (168) pode ser usada.

□

No cálculo do emparelhamento de Tate, o passo que exige maior esforço computacional é a determinação da função racional f tal que $(f) = r(P) - r(\mathcal{O})$. Dado que o emparelhamento de Weil este passo é repetido para a função g, é natural pensar-se que, com domínios compatíveis, o esforço computacional do emparelhamento de Weil seja, aproximadamente, o dobro do de Tate.

Normalmente r é um número primo com, pelo menos 160 bits de representação e, por isso, a exponenciação directa de uma função não é abordagem razoável. O **algoritmo de Miller** baseia-se na usual estratégia de cálculo eficiente de exponenciais por sucessivas do expoente e cálculo de quadrados.

Para este algoritmo recordemos o lema 197 (página 504), a função racional $\mu(P,Q)$ definida na noção 198 (página 505), e o seu divisor $(\mu(P,Q)) = (P) + (Q) - (P \oplus Q) - (\mathcal{O})$.

Para um ponto P da curva, seja $P_i = [i]P$ e considere-se a sucessão de funções racionais f_i tais que $f_1 = 1$ e

$$(P_i) - (\mathcal{O}) + (f_i) \;=\; i(P) - i(\mathcal{O}) \tag{169}$$

Note-se que, se $P \in E[r]$, $P_r = \mathcal{O}$ e que, portanto, f_r é a função que se pretende calcular. Por outro lado,

$$(\mu(P_i, P_j)) = (P_i) + (P_j) - (P_{i+j}) - (\mathcal{O}) \tag{170}$$

dado que $P_{i+j} = P_i \oplus P_j$.

216 LEMA *Com a sucessão de funções f_i verificando* (169) *verifica-se, a menos de uma constante multiplicativa,*

$$f_{i+j} \;=\; f_i \cdot f_j \cdot \mu(P_i, P_j) \tag{171}$$

Em particular

$$f_{2\,i} \;=\; f_i^2 \cdot \mu(P_i, P_i) \qquad \textit{e} \qquad f_{i+1} \;=\; f_i \cdot \mu(P_i, P) \tag{172}$$

Prova Dado que $\mu(P_i, P_j)$ verifica (170) tem-se $(f_i \cdot f_j \cdot \mu(P_i, P_j)) \;=\; (f_i) + (f_j) + (\mu(P_i, P_j)) \;=\; (f_{i+j})$. Como as funções racionais são determinadas, a menos de uma constante multiplicativa, pelos seus divisores, f_{i+j} é determinada por (171).

O algoritmo de Miller determina f_r usando sucessivamente as igualdades (172).

Objectivo Determinar $f_r(D)$ sendo $D = (Q \oplus S) - (S)$ para um S apropriado.

1. Fazer $f \leftarrow 1$, $T \leftarrow P$ e $n \leftarrow r$
2. Enquanto $n > 0$ executar repetidamente os seguintes passos
3. $T \leftarrow [2]T \quad ; \quad \lambda \leftarrow \mu(T,T) \quad ; \quad f \leftarrow f^2 \cdot \lambda(Q \oplus S)/\lambda(S)$
4. Se n é impar fazer $T \leftarrow T + P \quad ; \quad \lambda \leftarrow \mu(T,P) \quad ; \quad f \leftarrow f \cdot \lambda(Q \oplus S)/\lambda(S)$
5. $n \leftarrow n/2$.

O valor final de f contém $f_r(D)$; para calcular o emparelhamento de Tate basta agora determinar em K, o valor f^l usando um algoritmo de exponenciação eficiente.

8.7 Curvas Super-singulares e Implementação de Emparelhamentos

As condições para a definição do emparelhamento de Tate exigem uma escolha apropriada do corpo $K = \mathbb{F}_q(\boldsymbol{\mu}_r)$ impondo um garu de embebimento k pequeno, uma ordem r suficientemente grande e uma curva elíptica E/K cuja ordem seja um multiplo de r.

A verificação simultânea de todas estas condições é difícil se as curvas forem escolhidas arbitrariamente. Porém, para alguns tipos de curvas esta escolha é mais simples.

217 Noção
Seja $K = \mathbb{F}_q$ um corpo finito de característica p. Uma curva elíptica E/K é **super-singular** *quando satisfaz uma seguintes condições equivalentes:*

1. *Verifica-se $|E/K| = 1 \mod p$; equivalentemente $|E/K| = q + 1 - t$ para algum múltiplo t de p.*
2. *E/K não tem pontos de ordem p em $\overline{K}$; equivalentemente o grupo de torsão $E/\overline{K}[p]$ reduz-se a $\{\mathcal{O}\}$.*

Curvas que não verificam qualquer destas condições dizem-se **ordinárias**.

Para curvas arbitárias E/K (supersingulares ou ordinárias), o seguinte teorema ajuda a caracterizar o seu grau de embebimento.

218 Teorema (Waterhouse)
Nas condições da definição 217, para cada $a \in \mathbb{N}$, seja

$$T_a \;=\; \left\{\, t \in \mathbb{Z} \;\mid\; |E| = p^a + 1 - t \quad \textit{para alguma curva } E/K \,\right\} \tag{173}$$

Então, para todo $t \in T_a$, verifica-se $\gcd(t,p) \neq 1$, $|t| \leq 2\sqrt{p^q}$ e uma das seguintes condições

1. *a é par e $t = \pm 2\, p^{a/2}$*
2. *a é par, $(p \neq 1 \mod 3)$ e $t = \pm p^{a/2}$*
3. *a é impar, $p = 2, 3$ e $t = \pm p^{(a+1)/2}$*
4. *a é ímpar e $t = 0$*
5. *a é par, $(p \neq 1 \mod 4)$ e $t = 0$.*

Como consequência pode-se construir a seguinte tabela de curvas super-singulares para graus de embebimento $k = 1, 2, 3, 4, 6$. A tabela contém uma coluna q com o número de elementos de $K = \mathbb{F}_q$, a ordem $|E/K|$ da curva e a dimensão n tal que estrutura do grupo $E/\mathbb{F}_{q^k}$ é isomórfico com $\mathbb{Z}_n \times \mathbb{Z}_n$.

k	q	$\lvert E/K \rvert$	n
1	$p^{2\,b}$	$q \pm 2\sqrt{q} + 1$	$\sqrt{q} \pm 1$
2	$*$	$q + 1$	$q + 1$
3	$**$	$q + \sqrt{q} + 1$	$q^{3/2} - 1$
3	$**$	$q - \sqrt{q} + 1$	$q^{3/2} + 1$
4	$2^{2\,b+1}$	$q \pm \sqrt{2\,q} + 1$	$q^2 + 1$
6	$3^{2\,b+1}$	$q \pm \sqrt{3\,q} + 1$	$q^3 + 1$

O caso $(*)$ corresponde às entradas (4) e (5) no teorema 218. Os casos $(**)$ correspondem à entrada (2).