Le curve ellittiche e la loro applicazione alla crittografia

Alessio Palmero

17 ottobre 2006

Indice

Introduzione

La teoria delle curve ellittiche è uno dei crocevia fondamentali della matematica: vi si incontrano aritmetica, analisi, geometria e algebra; negli ultimi anni le sue applicazioni hanno raggiunto anche l'informatica. Inizialmente il loro studio era puramente astratto, come ad esempio nel campo della teoria dei numeri o in quello della geometria algebrica, fino ad arrivare alla dimostrazione del celeberrimo Ultimo Teorema di Fermat, una congettura sui numeri interi formulata nel 1600 dallo stesso Fermat in questo modo:

> *Cubum autem in duos cubos, aut quadratoquadratum in duos quadratoquadratos et generaliter nullam in infinitum ultra quadratum potestatem in duos eiusdem nominis fas est diuidere cuius rei demonstrationem mirabilem sane detexi. Hanc marginis exiguitas non caperet.*

In parole matematicamente più "moderne", questo significa che non esiste una terna di interi positivi x, y, z tali che $x^n + y^n = z^n$ con $n > 2$.

Il legame tra l'Ultimo Teorema di Fermat e le curve ellittiche risale al 1950, quando due matematici giapponesi, Yutaka Taniyama e Goro Shimura, congetturarono che ci fosse uno stretto legame tra un certo tipo di funzioni periodiche (dette forme modulari) e le curve ellittiche. Questa ipotesi, benché importante di per sé, inizialmente non catturò troppa attenzione nel mondo matematico finché, negli anni Ottanta, Frey e Ribet misero in luce che la congettura di Taniyama-Shimura implicava l'Ultimo Teorema di Fermat. Nel 1994, infine, Andrew Wiles dimostrò una parte della congettura di Taniyama-Shimura sufficiente per concludere positivamente questa secolare avventura matematica. Sempre nella teoria dei numeri, le curve ellittiche sono legate ai più importanti problemi della ricerca attuale, come la congettura "ABC" e la congettura di Birch e Swinnerton-Dyer. Nell'ambito delle sole curve ellittiche su campi finiti, dopo decenni di sviluppi gli studi hanno avuto come frutto tra gli altri la dimostrazione delle Congetture di Weil grazie ai lavori di Serre, Grothendieck e Deligne e della cosiddetta Ipotesi di Riemann per campi finiti.

Recentemente, invece, le curve ellittiche sono state studiate anche per risolvere molti problemi nell'ambito della crittografia, ultima nata tra le applicazioni pratiche della teoria dei numeri. Nonostante tracce di sistemi crittografici si ritrovino nell'antichità sin dal 1900 a. C., la nascita della crittografia moderna si può collocare durante la Seconda Guerra Mondiale, quando gli sforzi dei crittoanalisti alleati ebbero la meglio sul sistema tedesco basato sulla macchina Enigma. L'avvento dei calcolatori e dell'informatica teorica, a partire dai lavori di Turing, fece sì che si potessero sviluppare algoritmi basati su proprietà matematiche in precedenza intrattabili. Gli elementi fondamentali della crittografia al giorno d'oggi sono infatti i gruppi finiti con un numero di elementi dell'ordine di 10^{200} e i cosiddetti problemi

"NP" (non-deterministic polynomial), ovvero algoritmi molto difficili da calcolare a meno di avere un'informazione aggiuntiva. Per dare un'idea di cosa sia un problema NP, immaginiamo di dover preparare uno zaino, sapendo che quest'ultimo non può contenere più di n chilogrammi, altrimenti si rompe, e allo stesso tempo non può contenere meno di m chilogrammi, altrimenti facciamo una brutta figura con gli amici più muscolosi. Abbiamo inoltre svariati oggetti di diverso peso, tutti candidati a essere messi nello zaino. Il problema sta nel trovare un sottoinsieme di questi oggetti in modo tale che il loro peso totale sia compreso tra m e n. Se gli oggetti sono molti e di pesi molto diversi tra loro e m e n sono molto vicini (eventualmente coincidenti) si vede che non è semplice fare in modo che la somma dei singoli pesi degli oggetti dia esattamente come risultato un numero compreso tra m e n. Chiaramente, avendo a priori l'elenco degli oggetti del sottoinsieme, si verifica facilmente se la somma dei loro pesi cade o meno nell'intervallo richiesto.

Il problema NP utilizzato nel caso delle curve ellittiche riguarda invece i gruppi: data una di queste strutture possiamo, a partire da un elemento a dato, calcolare l'elemento $b = a + a + \cdots + a$ quante volte vogliamo, diciamo n. È invece spesso molto difficile, a partire da a e b appena descritti, trovare il corrispondente valore di n. Tuttavia, affinché il calcolo sia effettivamente impraticabile anche sui calcolatori attuali, è necessario che n sia molto grande e soprattutto che tutte le somme parziali $a + a$, $a + a + a$, e così via fino ad n, diano sempre elementi diversi: se così non fosse si creerebbe un ciclo ed esisterebbe inevitabilmente un $m < n$ tale che b sia m volte a. Questo può essere evitato prendendo un gruppo che sia allo stesso tempo molto "grande" (ovvero contenga molti elementi) e ciclico, ovvero in cui esiste almeno un elemento che sommato tante volte (fino al numero di elementi del gruppo) dia sempre risultati diversi; nonostante esistano gruppi di questo tipo "facili" da realizzare (come $\mathbb{Z}/p\mathbb{Z}$ con p primo), essi sono anche "facili" da risolvere dai calcolatori se p è troppo piccolo.

L'algoritmo probabilmente più noto tra quelli utilizzati comunemente è il cosiddetto RSA, un sistema di cifratura a chiave pubblica (per alcuni dettagli su questo tipo di sistemi rimandiamo alla sezione 1.2.3). Per mantenere un livello di sicurezza adeguato, l'RSA necessita (ad oggi) di una chiave lunga almeno 1024 bit. L'utilizzo delle curve ellittiche può dare un netto vantaggio proprio sotto questo punto di vista: i punti di queste curve formano un gruppo in cui, a parità del numero di elementi, la ricerca di n risulta sensibilmente più difficile, tanto che per avere un livello di sicurezza analogo a quello garantito dall'RSA con chiave di 1024 bit (circa 338 cifre decimali) è sufficiente utilizzare curve "buone" con chiavi lunghe solo 160 bit (circa 53 cifre decimali).

Questo testo vuole essere solamente un'introduzione ai sistemi di cifratura che possono essere implementati attraverso le curve ellittiche. Nel primo capitolo introdurremo il concetto di sistema di cifratura ed analizzeremo i vari tipi di sistemi crittografici, in particolare soffermandoci su quelli che si basano su gruppi; nel capitolo successivo descriveremo le curve ellittiche e mostreremo che esse possono essere viste come un gruppo; nel terzo capitolo, infine, sfrutteremo i risultati noti sulle curve ellittiche per analizzare i più importanti metodi di attacco contro i cifrari che si basano su di esse e vedremo come sia possibile scegliere curve che resistono molto bene a tutti gli attacchi descritti e sono quindi buone candidate per le applicazioni pratiche in crittografia.

1
La crittografia a chiave pubblica

In questo capitolo illustreremo rapidamente i concetti principali della crittografia, introducendo le tecniche matematiche necessarie, di cui parleremo più approfonditamente nei capitoli successivi.

1.1 Introduzione alla crittografia

L'obiettivo fondamentale della crittografia è consentire a due utenti, spesso soprannominati Alice e Bob, di comunicare su un canale potenzialmente insicuro senza permettere ad una terza persona (cui di solito viene dato il nome Eva) di comprendere il contenuto dei messaggi. Sono ad esempio canali insicuri il telefono oppure internet.

1.1.1 Definizioni

Il messaggio che Alice vuole spedire a Bob è detto **testo in chiaro**: per poter viaggiare attraverso un canale insicuro esso viene trasformato, attraverso un **sistema di cifratura** e una **chiave**, in un **crittogramma**. Chiave e sistema di cifratura non sono la stessa cosa: in un mondo dove le comunicazioni sono all'ordine del giorno, si deve supporre che il sistema di cifratura sia unico e di dominio pubblico, mentre la chiave è un parametro scelto dal mittente (o dal destinatario legittimo) che deve rimanere segreto. L'operazione che trasforma un testo in chiaro in un crittogramma viene detta **cifratura** (o **crittazione**) mentre l'operazione inversa è detta **decifrazione**. Si parla invece di **crittoanalisi** quando si vuole ricostruire il testo in chiaro a partire dal crittogramma senza avere a disposizione la chiave: quest'ultimo è il ruolo della spia (Eva nel nostro caso).

Definizione 1.1. *Un **sistema di cifratura**, o **cifrario**, è una* 5*-pla* $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, dove*

- $\mathcal{M}$ *è l'insieme di tutti i possibili messaggi in chiaro;*
- $\mathcal{C}$ *è l'insieme di tutti i possibili messaggi cifrati;*
- $\mathcal{K}$ *è l'insieme di tutte le possibili chiavi;*

- *Per ogni* $k \in \mathcal{K}$*, esistono una funzione* $e_k \in \mathcal{E}$ *e una funzione* $d_k \in \mathcal{D}$ *tali che* $e_k : \mathcal{M} \to \mathcal{C}$*,* $d_k : \mathcal{C} \to \mathcal{M}$ *e* $d_k(e_k(x)) = x$ *per ogni* $x \in \mathcal{M}$*.*

1.1.2 Il lavoro di Alice, Bob ed Eva

Alice e Bob scelgono una chiave da utilizzare. Con il termine scegliere intendiamo che Alice e Bob si siano messi d'accordo in qualche modo sulla chiave. Vedremo più avanti che esistono metodi per fare questo anche attraverso canali insicuri. Sia $k \in \mathcal{K}$ la chiave scelta al riparo dagli occhi indiscreti di Eva.

Sia $m \in \mathcal{M}$ il messaggio che Alice intende spedire a Bob. Esso sarà una stringa di caratteri alfabetici o numerici. Nella pratica la funzione e_k innanzi tutto divide m in modo che

$$m = m_1 m_2 \dots m_n$$

per qualche intero $n \geqslant 0$ e ogni m_i sia di lunghezza fissata. Ognuno di essi viene ora crittato usando un'opportuna funzione $\overline{e}_k$ relativa alla chiave k scelta. Alice calcola quindi $c_i = \overline{e}_k(m_i)$ e spedisce la stringa

$$c = e_k(m) = c_1 c_2 \dots c_n$$

a Bob. Quest'ultimo utilizzerà d_k in suo possesso per decifrare c ed ottenere il messaggio di partenza m. Se l'operazione è stata svolta in maniera corretta, Eva può venire in possesso solamente del crittogramma c: la sicurezza del messaggio m dipenderà dalla difficoltà per Eva di arrivare da c a m senza avere k.

1.2 Descrizione di alcuni cifrari

In questa sezione descriveremo in breve i sistemi di cifratura più utilizzati al giorno d'oggi. Essi si dividono in due categorie: a chiave privata (o simmetrici) e a chiave pubblica. Nella trattazione ci soffermeremo sui secondi, per i quali può essere utilizzata la teoria delle curve ellittiche dei capitoli successivi; rimangono tuttavia importanti anche i cifrari a chiave privata. Questi ultimi, infatti, sono decisamente più veloci e molto sicuri qualora le due chiavi siano in possesso unicamente del mittente e del destinatario legittimo. Una soluzione molto usata infatti prevede di utilizzare la chiave pubblica per trasmettere la chiave di un sistema simmetrico, da utilizzare per trasmettere successivamente la parte più cospicua di informazioni.

1.2.1 I cifrari a chiave privata (o simmetrici)

> *Alice si reca da un ferramenta e acquista un lucchetto, il quale è provvisto solo di due chiavi assolutamente identiche. Successivamente Alice incontra Bob e gli consegna una delle due chiavi. Alice scrive quindi un messaggio, lo introduce in una scatola, chiude tale scatola con il lucchetto e invia tutto a Bob, che chiaramente è l'unica persona che possa aprirla. Quest'ultimo risponde utilizzando lo stezzo procedimento. Ciò che può fare Eva è intercettare la scatola e impedire al destinatario di riceverla, ma non venire a conoscenza del contenuto.*

Il precedente esempio illustra in modo concreto come funziona un sistema simmetrico. Per poterlo utilizzare Alice e Bob devono prima accordarsi segretamente su una chiave $k \in \mathcal{K}$ che intendono utilizzare. Questo scambio deve avvenire in un modo sicuro, per esempio a voce in un luogo rumoroso, al riparo da orecchie indiscrete. A partire da quel momento Alice e Bob possono finalmente scambiarsi le informazioni in tutta tranquillità senza che Eva possa leggerle.

Nonostante questo sistema sia adeguato per molte applicazioni, esso presenta alcuni ostacoli che lo rendono difficilmente applicabile in determinati casi:

- potrebbe non esistere un canale sicuro attraverso cui comunicare la chiave da utilizzare;
- in una rete (come quella di internet) con n utenti, ogni possibile coppia di utenti deve possedere una chiave, per un totale di $n(n-1)/2$ chiavi, soluzione impraticabile per valori di n molto grandi;
- non è possibile ottenere una firma digitale da un sistema a chiave privata (vedere sezione 1.2.3).

Nel tentativo di risolvere questi inconvenienti, nel 1976 Diffie, Hellman e Merkle scoprirono la crittografia a chiave pubblica. Merkle si dedicò prevalentemente alla soluzione del secondo problema, mentre Diffie e Hellman spianarono la strada a generazioni future di matematici per la risoluzione simultanea di tutti e tre.

1.2.2 Lo scambio di chiavi di Diffie-Hellman

> *Alice prende una scatola, inserisce al suo interno un foglietto con il messaggio e chiude la scatola con un lucchetto che solo lei può aprire. Quindi spedisce il tutto a Bob, che apporrà alla scatola un ulteriore lucchetto, di cui solo lui ha la chiave. Rispedisce la scatola ad Alice, che toglie il primo lucchetto e invia nuovamente la scatola a Bob. Questo toglie il suo lucchetto e legge il messaggio. Anche in questo caso Eva, intercettando la scatola, non potrà leggerne il contenuto.*

Il questo sistema entrambe le parti (Alice e Bob) partecipano alla generazione di una chiave utilizzabile in un sistema a chiave privata.

Lo scambio di chiavi Diffie-Hellman si può implementare in questo modo:

- Alice e Bob scelgono un gruppo finito G e un elemento $\mathbf{g} \in G$, anche attraverso un canale insicuro;
- Alice quindi sceglie un intero casuale a, calcola $\mathbf{c} = \overbrace{\mathbf{g} + \mathbf{g} + \cdots + \mathbf{g}}^{a \text{ volte}} = a\mathbf{g}$ in G;
- Alice trasmette a Bob l'elemento $\mathbf{c}$ così ottenuto;
- Bob sceglie a sua volta un intero b, calcola $\mathbf{d} = b\mathbf{g}$ in G;
- Bob trasmette $\mathbf{d}$ ad Alice;
- Alice riceve $\mathbf{d}$ e calcola $a\mathbf{d} = a(b\mathbf{g}) = (ab)\mathbf{g}$;
- Bob riceve $\mathbf{c}$ e calcola $b\mathbf{c} = b(a\mathbf{g}) = (ba)\mathbf{g}$.

Chiaramente $(ab)\mathbf{g} = (ba)\mathbf{g}$ quindi Alice e Bob hanno in mano lo stesso elemento, tranquillamente utilizzabile come chiave.

Cosa succede se Eva intercetta i messaggi? Quest'ultima avrebbe in mano solamente G, $\mathbf{g}$, $a\mathbf{g}$ e $b\mathbf{g}$. Per poter ottenere $(ab)\mathbf{g}$ dovrebbe riuscire a calcolare a o b; questo problema (in genere "molto difficile") è noto come *problema del logaritmo discreto*[1], e verrà trattato a fondo successivamente.

1.2.3 La crittografia a chiave pubblica

> *Alice consegna a Bob un lucchetto aperto di cui solo lei possiede la chiave. In seguito Bob scrive il messaggio, lo introduce in una scatola, la chiude con il lucchetto e invia il tutto ad Alice, che può leggere il messaggio. Di nuovo Eva potrà intercettare il lucchetto aperto, la scatola chiusa, ma non avrà la possibilità di decifrare il messaggio.*

Tutti i sistemi finora trattati si basano sul concetto che un lucchetto è una funzione invertibile che trasforma una scatola chiusa in una aperta: mentre è molto semplice il problema diretto (chiudere la scatola), diventa quasi impossibile quello inverso (aprire la scatola), a meno di avere un'informazione aggiuntiva (la chiave). Con la chiave aprire la scatola diventa banale. L'idea centrale della crittografia a chiave pubblica è quella di separare chiave e lucchetto, in modo che nessuno (fuorché il legittimo destinatario) possa aprire il lucchetto, nemmeno chi l'ha chiuso.

Definizione 1.2. *Una **funzione unidirezionale** è una funzione $e : \mathcal{M} \to \mathcal{C}$ invertibile, tale che per ogni $m \in \mathcal{M}$ sia facile calcolare $c = e(m)$, mentre per ogni $c \in \mathcal{C}$ sia molto difficile calcolare $d(c)$ dove $d : \mathcal{C} \to \mathcal{M}$ è tale che $m = d(e(m))$.*

Più precisamente il problema diretto deve essere calcolabile in tempo "polinomiale", mentre il problema inverso deve richiedere (al minimo) un tempo "esponenziale", improponibile anche per il più veloce calcolatore. Questo significa che il numero di operazioni da svolgere, rispetto alle informazioni in possesso, deve essere rispettivamente un polinomio o un esponenziale. Tornando al problema del logaritmo discreto, si tratta di una funzione calcolabile in tempo esponenziale in cui il problema diretto (elevare alla n-esima potenza l'elemento di un gruppo) è evidentemente semplice.

Definizione 1.3. *Una funzione unidirezionale è detta **funzione trappola** se, ottenuta un'informazione aggiuntiva (detta **certificato**), può essere invertita facilmente, ovvero in tempo polinomiale.*

Rifacendoci alle notazioni della definizione 1.1, in un sistema crittografico a chiave pubblica l'algoritmo e_k è di pubblico dominio, mentre l'algoritmo d_k di decifrazione rimane segreto. Il certificato è la chiave k, attraverso cui si può trovare velocemente d_k. La difficoltà per Eva in questo caso risiede nell'ottenere d_k dato e_k.

Come possono Alice e Bob comunicare in sicurezza? Alice sceglie una chiave a e rende pubblico l'algoritmo e_a (chiave pubblica). Bob, analogamente, sceglie una chiave b e rende pubblico e_b. A questo

[1] In realtà il logaritmo farebbe pensare ad un gruppo moltiplicativo invece che ad uno additivo, ma è solo una questione formale. Nella nostra trattazione parleremo sempre di "somma" di elementi in G, in quanto la notazione additiva è più comune quando si parla del gruppo dei punti di una curva ellittica.

punto ogniqualvolta Alice vuole spedire un messaggio m_1 a Bob calcola $e_b(m_1)$. Bob può facilmente trovare d_b, in quanto possiede la chiave (certificato) b. Può poi rispondere ad Alice con un messaggio m_2 utilizzando la sua chiave pubblica e_a. Eva, invece, intercettando $e_b(m_1)$ e $e_a(m_2)$ senza i rispettivi a e b, si troverà sempre davanti ad un problema di tipo esponenziale. In compenso potrà utilizzare le chiavi pubbliche di Alice e Bob per comunicare con loro. Sta proprio in questo la potenza della chiave pubblica: ogni utente possiede una sua chiave pubblica, attraverso la quale chiunque può comunicare con lui in modo sicuro.

Vediamo ora alcuni sistemi di crittografia a chiave pubblica che utilizzano come funzione trappola il logaritmo discreto, di cui si è parlato nella sezione 1.2.2.

Il cifrario di ElGamal

L'algoritmo che segue è stato sviluppato da ElGamal nel 1985.

Innanzi tutto vengono scelti un gruppo finito G e un elemento $\mathbf{g} \in G$. Questi valori possono essere condivisi anche da vari utenti. La riservatezza sta nella scelta dell'intero a da parte del destinatario e dell'intero k da parte del mittente, come si vede dall'algoritmo sottostante. Il messaggio deve essere preventivamente trasformato in un elemento $\mathbf{m}$ di G attraverso un sistema su cui Alice e Bob si accordano (il metodo può essere anche di dominio pubblico).

- Alice sceglie un intero a e calcola $\mathbf{a} = a\mathbf{g}$; a è la chiave privata di Alice, $\mathbf{a}$ è la sua chiave pubblica.

Bob, per inviare un messaggio ad Alice, procede nel seguente modo:

- Bob sceglie un intero casuale k e calcola $k\mathbf{g}$ (ricordiamo che G e $\mathbf{g}$ sono pubblici);
- Bob calcola inoltre $k\mathbf{a}$ e successivamente $\mathbf{m} + k\mathbf{a}$ ($\mathbf{a}$ è la chiave pubblica di Alice);
- Bob invia la coppia $(k\mathbf{g}, \mathbf{m} + k\mathbf{a})$ ad Alice.

È da notare come la crittazione dipenda anche da un valore aleatorio k, scelto dal mittente di volta in volta e sconosciuto anche al destinatario legittimo. Questa particolarità incrementa la sicurezza del sistema: lo stesso messaggio, cifrato in tempi diversi, può dare luogo a crittogrammi diversi. Proprio per questo motivo la coppia $(k\mathbf{g}, \mathbf{m} + k\mathbf{a})$ sembra "ridondante" rispetto al messaggio: in questo modo Alice (destinatario legittimo) può avere informazioni su k attraverso il primo elemento $k\mathbf{g}$ e sul messaggio $\mathbf{m}$ grazie al secondo elemento $\mathbf{m} + k\mathbf{a}$.

Infine la decifrazione da parte di Alice avviene nel modo seguente:

- Alice somma $k\mathbf{g}$ un numero a di volte ottenendo $a(k\mathbf{g}) = k(a\mathbf{g}) = k\mathbf{a}$;
- Alice inverte facilmente l'elemento $k\mathbf{a}$ e trova $\mathbf{b} = -k\mathbf{a}$;
- infine Alice calcola $\mathbf{m}$ in questo modo: $\mathbf{m} = \mathbf{m} + k\mathbf{a} + \mathbf{b} = \mathbf{m} + k\mathbf{a} - k\mathbf{a} = \mathbf{m}$.

Come già osservato sopra, la presenza del parametro k da una parte aumenta il grado di sicurezza, ma dall'altro raddoppia la lunghezza del messaggio criptato. Affinché però il vantaggio sia reale, Bob deve cambiare spesso il valore di k, eventualmente ad ogni messaggio. Ovviamente questo non è un problema se a preoccuparsene è un computer.

La firma digitale di ElGamal

La firma digitale non serve per nascondere un messaggio, ma per permettere a chiunque (destinatario, ma anche terze parti) di verificare l'autenticità di esso. Deve in pratica sostituire la firma cartacea, quindi:

- il destinatario deve essere sicuro che il messaggio derivi proprio da quel mittente;
- il destinatario deve essere sicuro che il messaggio non sia stato modificato durante il percorso;
- il mittente non può ripudiare il messaggio inviato.

Prima di descrivere il funzionamento della firma digitale di ElGamal abbiamo bisogno di un concetto molto importante: la funzione hash. Il nostro obiettivo è trovare una stringa (da allegare al messaggio) che rispetti i punti sopra elencati. Essa deve dipendere dal messaggio, dal mittente, ed essere fatta in modo che due messaggi (o due mittenti) diversi diano luogo a stringhe diverse. Si potrebbe pensare che inevitabilmente una funzione di questo tipo produca una stringa lunga quanto il messaggio (come proposto inizialmente da ElGamal). In realtà, data la scarsa quantità di messaggi sensati rispetto a tutti i messaggi possibili (quelli quindi con tutte le lettere possibili in tutti gli ordini possibili), una funzione hash può firmare un messaggio di qualsiasi lunghezza attraverso una stringa di lunghezza fissata con un ottimo grado di sicurezza. Il nostro compito è quindi quello di trovare una funzione che, a partire da un messaggio m, restituisca una stringa di lunghezza fissata t.

Definizione 1.4. *Sia $\mathcal{M}_t$ lo spazio delle stringhe (messaggi) di lunghezza t, allora si definisce **funzione hash** una funzione $h : \mathcal{M} \to \mathcal{M}_t$ che soddisfi alle seguenti caratteristiche:*

1. *dato un messaggio di qualsiasi lunghezza, la funzione fornisce una stringa di lunghezza t;*
2. *deve essere computazionalmente impossibile trovare due messaggi che mappano nella stessa stringa;*
3. *deve essere computazionalmente impossibile trovare un messaggio che mappa in una stringa prestabilita;*
4. *deve essere computazionalmente impossibile risalire al messaggio m dal valore di $h(m)$.*

L'algoritmo proposto da ElGamal si basa sempre sul problema del logaritmo discreto. Immaginiamo nuovamente che Alice debba spedire un messaggio a Bob. Vengono scelti un gruppo finito G, un elemento $\mathbf{g} \in G$ e una funzione hash $f : G \to \mathbb{Z}$. Sia n l'ordine di $\mathbf{g}$. La parte m_i del messaggio (vedere sezione 1.1.2) deve essere preventivamente trasformata in un numero intero m minore di n.

- Alice sceglie un intero a e calcola $\mathbf{a} = a\mathbf{g}$; a è la sua chiave privata, $\mathbf{a}$ è la sua chiave pubblica;
- Alice sceglie ora un intero casuale k, diverso per ogni messaggio, tale che $\text{MCD}(k, n) = 1$ e calcola $\mathbf{r} = k\mathbf{g}$;
- calcola l'inverso k^{-1} che esiste per la scelta di k;
- infine calcola $s \equiv k^{-1}(m - af(\mathbf{r})) \bmod n$;

- Il messaggio firmato è $(m, \mathbf{r}, s)$.

Si noti che nel caso di un gruppo formato da elementi che non sono numeri (come quello dei punti di una curva ellittica), la terna non è formata da interi, bensì da due interi m e s e dall'elemento $\mathbf{r}$ del gruppo G. Bob può ora verificare la firma del documento.

- Calcola $\mathbf{u} = f(\mathbf{r})\mathbf{a} + s\mathbf{r}$ e $\mathbf{v} = m\mathbf{g}$.
- Se $\mathbf{u} = \mathbf{v}$ dichiara la firma valida.

Per verificare che il procedimento funziona, osserviamo che

$$sk = kk^{-1}(m - af(\mathbf{r})) + zn = m - af(\mathbf{r}) + zn$$

per qualche z e quindi

$$sk\mathbf{g} = (m - af(\mathbf{r}))\mathbf{g} + zn\mathbf{g} = (m - af(\mathbf{r}))\mathbf{g} + z\mathbf{0} = (m - af(\mathbf{r}))\mathbf{g}.$$

Ora si vede facilmente che $\mathbf{u} = \mathbf{v}$:

$$\mathbf{u} = f(\mathbf{r})\mathbf{a} + s\mathbf{r} = f(\mathbf{r})a\mathbf{g} + sk\mathbf{g} = af(\mathbf{r})\mathbf{g} + (m - af(\mathbf{r}))\mathbf{g} = m\mathbf{g} = \mathbf{v}.$$

Se Eva intercettasse il messaggio e sapesse calcolare il logaritmo discreto, sarebbe in grado di ricavare a da $\mathbf{a}$ e $\mathbf{g}$. In questo caso potrebbe apporre la firma di Alice su qualsiasi messaggio. Alternativamente, conoscendo k potrebbe sfruttare la relazione $sk = kk^{-1}(m - af(\mathbf{r})) \bmod n$ allo stesso modo. Alice deve quindi tenere segreti a e k; soprattutto deve cambiare la scelta di k per ogni firma. Supponiamo infatti che Alice utilizzi due volte lo stesso valore di k: Eva nota immediatamente che il valore di $\mathbf{r}$ nei due messaggi è lo stesso, e che quindi anche k deve essere uguale. Detti $(m_1, \mathbf{r}, s_1)$ e $(m_2, \mathbf{r}, s_2)$ i due messaggi firmati, allora dalle relazioni

$$\begin{aligned} s_1 &\equiv k^{-1}(m_1 - af(\mathbf{r})) \bmod n \\ s_2 &\equiv k^{-1}(m_2 - af(\mathbf{r})) \bmod n \end{aligned}$$

si ottiene

$$(s_1 - s_2)k \equiv (m_1 - m_2) \bmod n.$$

Se $s_1 - s_2$ è primo con n, allora k è univocamente determinato. Se invece così non fosse, il ragionamento sarebbe comunque possibile, anche se un po' più complesso. Sia $d = \mathrm{MCD}(s_1 - s_2, n)$, si pongono

$$m' = \frac{m_1 - m_2}{d}, \quad s' = \frac{s_1 - s_2}{d}, \quad n' = \frac{n}{d}.$$

Eva scopre che $m' \equiv ks' \bmod n'$. Ci sono quindi d possibili valori per k, distanti tra loro n': quello giusto sarà l'unico che soddisfa la relazione $\mathbf{r} = k\mathbf{g}$. Una volta ottenuto k, Eva calcola velocemente a dalla

relazione $k_s \equiv (m - af(\mathbf{r})) \bmod n$ se $\text{MCD}(n, f(\mathbf{r})) = 1$; in caso contrario trova comunque $\text{MCD}(n, f(\mathbf{r}))$ valori e prende come a l'unico valore che soddisfa $\mathbf{a} = a\mathbf{g}$.

2

Le curve ellittiche e la loro legge di gruppo

Da quello che abbiamo appena detto, per poter cifrare un documento abbiamo bisogno di un gruppo finito. Vediamo ora alcune tipologie di curve ellittiche particolarmente adatte al nostro scopo.

2.1 Le curve ellittiche

Ricordiamo alcune definizioni importanti.

Definizione 2.1. *Dato un campo* K*, il* ***piano proiettivo*** $\mathbb{P}^2(K)$ *è l'insieme delle classi di equivalenza della relazione* $\sim$ *su* $K^3\backslash\{(0,0,0)\}$*, dove* $(x_1,x_2,x_3) \sim (y_1,y_2,y_3)$ *se e solo se esiste* $\lambda \in K^*$ *tale che* $(x_1,x_2,x_3) \sim (\lambda y_1, \lambda y_2, \lambda y_3)$*. La classe di equivalenza di* (x_1,x_2,x_3) *si denota con* $(x_1 : x_2 : x_3)$*. Osserviamo che la scrittura* $(0:0:0)$ *non corrisponde ad un elemento di* $\mathbb{P}^2(K)$*.*

Per il nostro scopo, con un abuso di notazioni, possiamo considerare, dato un campo K, lo **spazio affine** $\mathbb{A}^2(K)$ come il prodotto cartesiano $K \times K$.

Definizione 2.2. *Sia* K *un campo. Un'equazione della forma*

$$F(x,y,z) = y^2z + a_1xyz + a_3yz^2 - x^3 - a_2x^2z - a_4xz^2 - a_6z^3 = 0 \tag{2.1}$$

dove i parametri $a_1, \cdots, a_6$ *e le variabili* x, y, z *sono elementi del campo* K *è detta* ***equazione di Weierstrass****.*

Concludiamo la carrellata con la definizione di curva ellittica.

Definizione 2.3. *Dato un campo* K*, una* ***curva ellittica*** E/K *è l'insieme delle soluzioni in* $\mathbb{P}^2(K)$ *di una equazione di Weierstrass.*

Essendo la (2.1) un polinomio omogeneo di terzo grado, la definizione non porta a contraddizioni, in quanto se un punto (x,y,z) soddisfa la (2.1), allora anche il punto $\lambda(x,y,z)$ con $\lambda \neq 0$ godrà della stessa proprietà. Abbiamo quindi appena verificato che le curve ellittiche sono ben definite.

Se ci limitiamo al caso $z \neq 0$, per identificare un punto in $\mathbb{P}^2(K)$, la terza coordinata z è ridondante; per la relazione di equivalenza ereditata dallo spazio proiettivo possiamo sempre moltiplicare un punto (x, y, z) per $1/z$ ottenendo il punto equivalente $(x/z, y/z, 1)$. Questo ci permette di semplificare l'equazione (2.1) attraverso la sostituzione (per $z \neq 0$)

$$(x, y, z) \longmapsto (x/z, y/z, 1)$$

che ci porta alla nuova equazione (detta equazione di Weierstrass non omogenea)

$$f(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0 \tag{2.2}$$

dove abbiamo ribattezzato x/z e y/z rispettivamente x e y.

Per $z \neq 0$, quindi, le soluzioni in $\mathbb{P}^2(K)$ dell'equazione di Weierstrass (2.1) possono essere messe in corrispondenza biunivoca con le soluzioni in $\mathbb{A}^2(K)$ della (2.2).

Cosa succede se $z = 0$? Azzerando la terza coordinata nell'equazione (2.1) troviamo $x^3 = 0$. Tutti i punti ottenuti da questa operazione sono quelli dell'insieme $(0, y, 0)$ al variare di y, con $y \neq 0$, il che vuol dire (data la definizione di spazio proiettivo) che tale insieme è composto dall'unico elemento $(0 : 1 : 0)$.

Definizione 2.4. *Sia data una curva ellittica* E/K. *Il punto* $(0 : 1 : 0)$ *è detto* ***punto all'infinito*** *della curva e si indica con* $\mathcal{O}$.

Possiamo quindi fornire una nuova definizione (più utile per i calcoli) di curva ellittica: dato un campo K, una curva ellittica E/K è in pratica l'insieme delle soluzioni in $\mathbb{A}^2(K)$ dell'equazione (2.2) più il punto all'infinito $\mathcal{O}$ della definizione 2.4.

2.2 Curve singolari

Come già visto, una curva ellittica E/K è l'insieme delle soluzioni di una funzione del tipo (2.1) oppure (2.2). Essendo di terzo grado, ci si aspetta che una retta incontri la curva in tre punti (contati con le rispettive molteplicità). Vediamo come evitare curve "anomale", cioè in cui la tangente a qualche punto non è ben definita: questo fatto impedirebbe la legge di gruppo della sezione 2.3.

Definizione 2.5. *Sia* E/K *una curva di equazione (2.1). Un punto* $P = (x, y)$ *si dice* ***punto singolare*** *se e solo se*

$$\frac{\partial F}{\partial x}(P) = \frac{\partial F}{\partial y}(P) = \frac{\partial F}{\partial z}(P) = 0.$$

La curva E/K *si dice* ***curva non singolare*** *se non ha punti singolari.*

Silverman, in [25], ci assicura che una "scelta aleatoria" dei coefficienti $a_1, \ldots, a_6$ produce con buona probabilità una curva non singolare. La figura 2.1 mostra due esempi di curve singolari. In tutto il resto della trattazione, se non viene detto esplicitamente, prenderemo in considerazione solamente curve ellittiche non singolari.

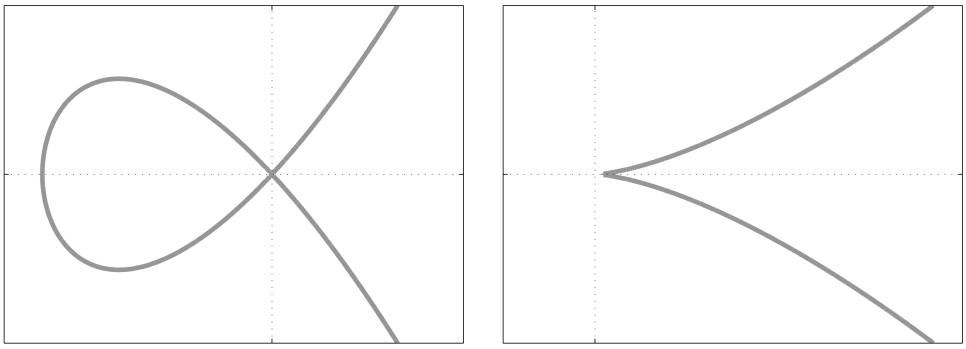

Figura 2.1: Due esempi di curve singolari.

2.3 La legge di gruppo

Sia E/K una curva ellittica. Essa avrà equazione (2.1); se ad esempio $K = \mathbb{R}$ e il suo grafico sarà composto da una o da due componenti. Nella figura 2.2 sono rappresentate le due tipologie di grafico a meno di rotazioni, riflessioni o traslazioni.

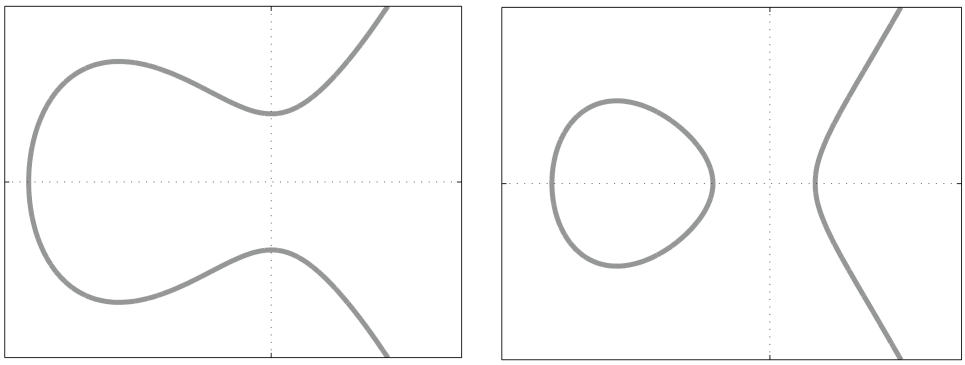

Figura 2.2: Le due tipologie di grafico di una curva ellittica non singolare.

Abbiamo finora parlato di gruppo dei punti di una curva ellittica. Per introdurre una struttura di questo tipo abbiamo bisogno di un'operazione.

Sia L una retta in $\mathbb{P}^2(K)$. Se L interseca la curva in due punti, allora sicuramente la intersecherà anche in un terzo punto (due o tre dei punti in questione potrebbero essere coincidenti, nel qual caso L risulterebbe tangente a E/K).

Definizione 2.6. *Sia E/K una curva ellittica e sia A un punto fissato di E/K. Siano ora $P, Q \in E$, L la retta che congiunge P con Q, e R il terzo punto di intersezione di E con L. Siano ora L' la retta passante per R e A, e R' il terzo punto di intersezione tra E ed L'. Allora R' è la **somma** $\mathbf{P} \oplus \mathbf{Q}$ (si veda la figura 2.3).*

La scelta del punto A è totalmente arbitraria. In crittografia si usa prendere $A = \mathcal{O}$: in questo modo

R' è esattamente il punto speculare di R rispetto all'asse di simmetria della curva, il che diminuisce notevolmente il carico di lavoro di un eventuale elaboratore (per uno studio nel caso generale si veda [6]).

Teorema 2.7. *Una curva ellittica* E/K *non singolare è un gruppo abeliano rispetto all'operazione* $\oplus$ *con elemento neutro* $\mathcal{O}$.

Dimostrazione. Verifichiamo che la definizione 2.6 soddisfa gli assiomi di gruppo.

- L'operazione è chiusa, come vedremo nella sezione 2.3.1. Dati due punti su una curva ellittica le operazioni che si devono eseguire per calcolare le coordinate del terzo punto sono solamente addizioni e moltiplicazioni nel campo K. Sono quindi anch'esse in K.
- L'elemento neutro è $\mathcal{O}$. Dato un punto P, proviamo a calcolare $P + \mathcal{O}$. La retta passante per P e per $\mathcal{O}$ incontra la curva nell'unico altro punto P' avente la stessa ascissa di P. La retta passante per P' e $\mathcal{O}$ chiaramente incontra la curva in P, da cui $P + \mathcal{O} = P$.
- Riferendoci alla figura 2.2, l'inverso di un punto P sulla curva è l'unico altro punto P' avente la stessa ascissa di P.
- La somma è commutativa, in quanto la retta che passa per P e Q è la stessa che passa per Q e P.
- La prova della proprietà associativa è lunga e richiede nozioni di algebra che esulano dagli obiettivi di questo testo. Per una dimostrazione completa si veda [29]. □

Che caratteristiche possiede il gruppo appena descritto? Se il campo K su cui viene costruita la curva è formato da un numero finito di elementi, anche il gruppo godrà della medesima proprietà: in questo modo può essere utilizzato per implementare i cifrari del primo capitolo.

Concludiamo questa sezione con un teorema che fornisce la struttura del gruppo.

Teorema 2.8. *Sia* E/K *una curva ellittica e sia* q *la cardinalità di* K. *Allora il gruppo formato dai suoi punti è isomorfo a* $\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$[1] *dove* $n_1 | n_2$ *e* $n_2 | q - 1$.

2.3.1 Sommare i punti

Vediamo come si sommano i punti nella pratica.

Siano $P(x_1, y_1)$ e $Q(x_2, y_2)$ due punti di una curva ellittica E/K. Siano (x_3, y_3) le coordinate di $P + Q$ e sia L la retta passante per P e Q (se $P \neq Q$) oppure tangente a P (nel caso in cui $P = Q$). La pendenza di L sarà

$$m = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{se } P \neq Q \\ \dfrac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} & \text{se } P = Q. \end{cases}$$

Se $q = y_1 - mx_1$, allora l'equazione di l è $y = mx + q$. Per trovare il terzo punto di intersezione scriviamo il sistema retta–curva, e lo risolviamo sfruttando il fatto che abbiamo già due soluzioni: i punti P e Q.

[1] Per comodità di notazione si indicherà con $\mathbb{Z}_n$ l'insieme $\mathbb{Z}/n\mathbb{Z}$

Sostituendo $y = mx + q$ nella (2.2) otteniamo

$$x^3 + a_2x^2 + a_4x - (mx + q)^2 - a_1x(mx + q) - a_3(mx + q) = 0 \tag{2.3}$$

le cui radici sono proprio x_1, x_2 (che abbiamo già) e x_3. Quindi la (2.3) può essere scritta come

$$(x - x_1)(x - x_2)(x - x_3) = x^3 - (x_1 + x_2 + x_3)x^2 + \cdots = 0. \tag{2.4}$$

Uguagliando i termini in x^2 della (2.3) e della (2.4) otteniamo

$$-(x_1 + x_2 + x_3) = a_2 - m^2 - a_1m$$

da cui

$$x_3 = m^2 + a_1m - a_2 - x_1 - x_2 \tag{2.5}$$
$$y_3 = -(m + a_1)x_3 - q - a_3. \tag{2.6}$$

Se $P, Q \in E/K$ allora il calcolo $P + Q$ avviene utilizzando solamente addizioni e moltiplicazioni. Se K è un campo finito questo vuol dire che il calcolo di $P + Q$ avviene in tempo polinomiale.

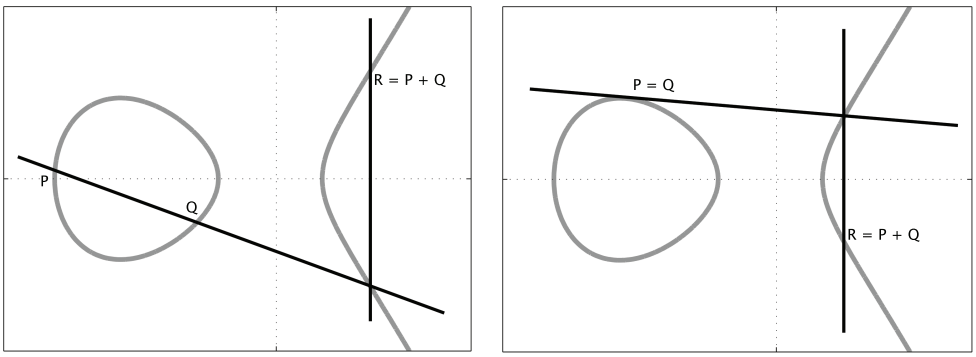

Figura 2.3: Come si sommano i punti (caso incidente e caso tangente).

2.4 Isomorfismo tra curve

Definiamo i seguenti valori

$$d_2 = a_1^2 + 4a_2$$
$$d_4 = 2a_4 + a_1a_3$$
$$d_6 = a_3^2 + 4a_6$$
$$d_8 = a_6d_2 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

$$\begin{aligned} c_4 &= d_2^2 - 24d_4 \\ \Delta &= -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2 d_4 d_6 \quad (2.7) \\ j(E) &= c_4^3/\Delta. \quad (2.8) \end{aligned}$$

Definizione 2.9. *Data un'equazione di Weierstrass (2.2), la quantità Δ espressa dalla (2.7) si chiama* ***discriminante*** *della cubica.*

Definizione 2.10. *Data un'equazione di Weierstrass (2.2), la quantità $j(E)$ espressa dalla (2.8) si chiama* ***j-invariante*** *della cubica.*

Proposizione 2.11. *Una curva ellittica è non singolare se e solo se il discriminante della sua equazione di Weierstrass è non nullo.*

Definizione 2.12. *Siano E_1 e $E_2 \subseteq \mathbb{P}^2(K)$ due curve ellittiche. Una* ***mappa razionale da*** E_1 ***a*** E_2 *è un'applicazione della forma:*

$$\phi : E_1 \to E_2$$

$$\phi = [f_0, f_1, f_2]$$

dove $f_i = \frac{h_i}{k_i}$ sono quozienti di polinomi omogenei con k_i non identicamente nullo su E_1 e hanno la proprietà che, per ogni punto $P \in E_1$ in cui i k_i sono tutte diverse da 0,

$$\phi(P) = [f_0(P), f_1(P), f_2(P)] \in E_2.$$

Definizione 2.13. *Una mappa razionale ϕ è detta* ***regolare in*** $P \in E_1$ *se esiste una funzione $g = \frac{h}{k}$ con h e k come sopra tale che*

1. *ogni gf_i ha denominatore che non si annulla in P;*
2. *esiste i tale che $(gf_i)(P) \neq 0$.*

Definizione 2.14. *Una mappa razionale regolare in ogni punto è detta* ***morfismo****.*

Queste ultime tre definizioni si applicano anche nel caso di varietà algebriche proiettive arbitrarie.

Definizione 2.15. *Due curve ellittiche E/K ed F/K si dicono* ***isomorfe*** *se esistono due morfismi $\phi : E/K \to F/K$ e $\psi : F/K \to E/K$ tali che $\psi \circ \phi$ e $\phi \circ \psi$ siano le funzioni identiche rispettivamente in E/K e F/K. L'isomorfismo si denota con il simbolo $\simeq$.*

Teorema 2.16. *Due curve E/K e F/K date dalle equazioni*

$$\begin{aligned} E : y^2 + a_1xy + a_3y &= x^3 + a_2x^2 + a_4x + a_6 \quad (2.9) \\ F : y^2 + \overline{a}_1xy + \overline{a}_3y &= x^3 + \overline{a}_2x^2 + \overline{a}_4x + \overline{a}_6 \quad (2.10) \end{aligned}$$

sono isomorfe su K se e solo se esistono $u, r, s, t \in K, u \neq 0$, tali che il cambio di variabile

$$(x, y) \longrightarrow (u^2x + r, u^3y + u^2sx_t) \quad (2.11)$$

trasforma l'equazione E *nell'equazione* F. *La relazione* $\simeq$ *è una relazione di equivalenza.*

Sostituendo la trasformazione (2.11) in (2.9) ed uguagliando i coefficienti ottenuti con quelli di (2.10) possiamo vedere come variano. Questo ci porta al seguente insieme di equazioni:

$$\begin{aligned} u\overline{a}_1 &= a_1 + 2s \\ u^2\overline{a}_2 &= a_2 - sa_1 + 3r - s^2 \\ u^3\overline{a}_3 &= a_3 + ra_1 + 2t \\ u^4\overline{a}_4 &= a_4 - sa_3 + 2ra_2 - (t + rs)a_1 + 3r^2 - 2st \\ u^6\overline{a}_6 &= a_6 + ra_4 + r^2a_2 + r^3 - ta_3 - t^2 - rta_1 \end{aligned} \tag{2.12}$$

Le equazioni (2.12) spiegano la strana numerazione dei coefficienti $a_1, \ldots, a_6$.

Il teorema 2.16 equivale quindi al seguente teorema.

Teorema 2.17. *Due curve ellittiche su* K *sono isomorfe se e solo se esistono* $u, r, s, t \in K, u \neq 0$ *che soddisfano le equazioni (2.12).*

Un altro metodo per verificare se due curve sono isomorfe su un campo K risiede nel seguente teorema.

Teorema 2.18. *Se due curve ellittiche* E/K *e* F/K *sono isomorfe su* K*, allora* $j(E) = j(F)$. *Se* K *è algebricamente chiuso, allora vale anche il viceversa.*

Perché in crittografia è importante sapere quando due curve sono isomorfe? Un metodo per incrementare il livello di sicurezza di un sistema crittografico a chiave pubblica è quello di cambiare di tanto in tanto il gruppo su cui si opera. Il seguente teorema ci dice che per cambiare il gruppo sui punti di una curva ellittica non è sufficiente cambiare la curva.

Teorema 2.19. *Se due curve ellittiche sono isomorfe, allora sono isomorfi anche i gruppi da esse generati.*

Non vale il viceversa, come mostra il seguente esempio.

Si prenda $K = \mathbb{F}_4 = \{0, 1, c_1, c_2\}$. Le seguenti tabelle mostrano come si comportano gli elementi del campo con le due operazioni $\times$ e $+$.

$+$	0	1	c_1	c_2
0	0	1	c_1	c_2
1	1	0	c_2	c_1
c_1	c_1	c_2	0	1
c_2	c_2	c_1	1	0

$\times$	0	1	c_1	c_2
0	0	0	0	0
1	0	1	c_1	c_2
c_1	0	c_1	c_2	1
c_2	0	c_2	1	c_1

Si considerino le due curve definite su $\mathbb{F}_4$

$$\begin{aligned} E/K : y^2 + xy &= x^3 + c_1x^2 + c_1 \\ F/K : y^2 + xy &= x^3 + c_1x^2 + c_2. \end{aligned}$$

Le equazioni (2.12) diventano allora (ricordiamo che la caratteristica di $\mathbb{F}_4$ è 2)

$$\begin{aligned} u\overline{a}_1 &= a_1 \\ u^2\overline{a}_2 &= a_2 + sa_1 + r + s^2 \\ \overline{a}_3 &= a_3 + ra_1 \\ u\overline{a}_4 &= a_4 + sa_3 + (t + rs)a_1 + r^2 \\ \overline{a}_6 &= a_6 + ra_4 + r^2a_2 + r^3 + ta_3 - t^2 + rta_1. \end{aligned}$$

Sostituendo i valori $a_1, \ldots, a_6$ delle equazioni di E/K e F/K otteniamo

$$\begin{aligned} u &= 1 \\ u^2c_1 &= c_1 + s + r + s^2 \\ 0 &= r \\ 0 &= t + rs + r^2 \\ c_1 &= c_2 + r^2c_1 + r^3 + t^2 + rt. \end{aligned}$$

Si vede immediatamente che $u = 1$, $r = 0$, e di conseguenza per la quarta equazione $t = 0$. Sostituendo questi valori nell'ultima equazione si giunge a $c_1 = c_2$, assurdo: quindi le due curve non sono isomorfe.

Vediamo ora come sono costituiti i rispettivi gruppi.

x	E/K	Punti	F/K	Punti
0	$y^2 = c_1$	$(0, c_2)$	$y^2 = c_2$	$(0, c_1)$
1	$y^2 + y = 1$	$(1, c_1), (1, c_2)$	$y^2 + y = 0$	$(1, 1), (1, 0)$
c_1	$y^2 + c_1y = c_1$		$y^2 + c_1y = c_2$	$(c_1, 1), (c_1, c_2)$
c_2	$y^2 + c_2y = 0$	$(c_2, 0), (c_2, c_2)$	$y^2 + c_2y = 1$	

I due gruppi hanno quindi entrambi sei punti (i cinque appena trovati insieme con $\mathcal{O}$).

Poiché l'ordine di ogni elemento divide la cardinalità del gruppo, se facciamo vedere che vale $4P \neq P$ per almeno un $P \in E/K$ o $P \in F/K$, allora avremo dimostrato che il gruppo è isomorfo al gruppo ciclico C_6. Essendo il campo di caratteristica 2, le formule per raddoppiare un punto vengono facilitate (si noti che le formule non dipendono da a_6, quindi possono essere utilizzate per entrambe le curve). Nel nostro caso:

$$\begin{aligned} m &= \frac{x^2 + y}{x} \\ q &= mx + y = x^2 \\ x_s &= m^2 + m + c_1 \\ y_s &= mx_s + x_s + q \end{aligned}$$

Prendiamo $(1, c_1) \in E/K$ e $(1, 0) \in F/K$. Avremo

$$\begin{aligned} 4(1, c_1) &= 2\big(2(1, c_1)\big) = 2(c_2, 0) = (c_2, c_2) \\ 4(1, 0) &= 2\big(2(1, 0)\big) = 2(c_1, 1) = (c_1, c_2), \end{aligned}$$

quindi entrambi i gruppi sono isomorfi a C_6.

3

Il logaritmo discreto

3.1 L'accoppiamento di Weil

Definizione 3.1. *Sia* E/K *una curva ellittica e sia* n *un numero positivo. Un punto* P *è detto* **punto di n-torsione se** $nP = \mathcal{O}$**.**

Riferendoci alla figura 2.2, vediamo che ad esempio i 2-punti di torsione sono tutti e soli i punti con tangente verticale (oltre ovviamente il punto $\mathcal{O}$).

Chiaramente, se un punto è un punto di n-torsione, ogni suo multiplo mantiene la stessa proprietà. L'insieme $E[n]$ definito da

$$E[n] = \{P \in E/\overline{K} \text{ tale che } nP = \mathcal{O}\}$$

può essere quindi visto come gruppo astratto degli n-punti di torsione di E/K.

Il teorema seguente fornisce la struttura del gruppo degli n-punti di torsione.

Teorema 3.2. *Sia* E/K *una curva ellittica e sia* n *un numero intero positivo. Sia inoltre* p *la caratteristica di* K.

- *Se* $p \nmid n$ *oppure* $p = 0$*, allora*
$$E[n] \simeq \mathbb{Z}_n \oplus \mathbb{Z}_n.$$
- *Se* $p|n$*, sia* $n = p^r n'$ *con* $p \nmid n'$*. Allora*
$$E[n] \simeq \mathbb{Z}_{n'} \oplus \mathbb{Z}_{n'} \quad \text{oppure} \quad E[n] \simeq \mathbb{Z}_n \oplus \mathbb{Z}_{n'}.$$

Il secondo punto di questo teorema ci permette di dividere le curve ellittiche utili in crittografia in due categorie.

Definizione 3.3. *Una curva ellittica* E *definita su un campo* K *di caratteristica* p *è detta* ***supersingolare*** *se* $E[p] \simeq 0$*. Altrimenti la curva è detta* ***ordinaria****.*

Proposizione 3.4. *Una curva ellittica* E *su un campo* $\mathbb{F}_{p^m}$ *di cardinalità* $p^m + 1 - a$ *con* $a \equiv 0 \bmod p$ *è supersingolare.*

Se n è un numero intero positivo primo con p, possiamo scegliere una base $\{b_1, b_2\}$ per $E[n] \simeq \mathbb{Z}_n \oplus \mathbb{Z}_n$. Questo significa che ogni elemento di $E[n]$ può essere scritto come $m_1 b_1 + m_2 b_2$ con m_1 e m_2 unicamente determinati modulo n. Sia ora

$$\mu_n = \{x \in \overline{K} \text{ tale che } x^n = 1\}$$

il gruppo delle n-esime radici unitarie in $\overline{K}$. Poiché p non divide n, l'equazione $x^n = 1$ ha esattamente n radici in $\overline{K}$. Il gruppo μ_n è ciclico di ordine n e i suoi generatori sono detti radici primitive n-esime dell'unità. Quindi se ζ è un generatore di μ_n allora $\zeta^k = 1$ se e solo se n divide k.

Teorema 3.5. *Sia* E/K *una curva ellittica e* n *un numero intero positivo. Si supponga che la caratteristica* p *di* K *non divida* n*. Allora esiste una funzione*

$$e_n : E[n] \times E[n] \to \mu_n$$

che soddisfa alle seguenti proprietà:

- e_n *è bilineare;*
- e_n *è non degenere;*
- $e_n(T, T) = 1$ *per ogni* $T \in E[n]$*;*
- $e_n(S, T) = e_n(T, S)^{-1}$ *per ogni* $S, T \in E[n]$*.*
- $e_n(\sigma S, \sigma T) = \sigma e_n(S, T)$ *per ogni* $S, T \in E[n]$ *e per ogni automorfismo* σ *di* $\overline{K}$ *tale che* σ *è la mappa identica sui coefficienti di* E*.*

Definizione 3.6. *La funzione* e_n *del teorema 3.5 viene detta* **accoppiamento di Weil.**

Corollario 3.7. *Se* $E[n] \subseteq E/K$*, allora* $\mu_n \subset K$*.*

3.2 La mappa di Frobenius

Definizione 3.8. *Sia* $\mathbb{F}_q$ *un campo finito. La funzione* $\phi_q : \overline{\mathbb{F}}_q \to \overline{\mathbb{F}}_q$ *definita da*

$$\phi_q(x) = x^q \text{ per ogni } x \in \overline{\mathbb{F}}_q$$

è detta ***mappa di Frobenius*** *alla potenza* q*.*

Proposizione 3.9. *Sia* p *un numero primo e* $q = p^n$*. Allora valgono le seguenti:*

1. $\overline{\mathbb{F}}_q = \overline{\mathbb{F}}_p$*;*

2. ϕ_q *è un automorfismo di* $\overline{\mathbb{F}}_q$*, valgono cioè per ogni* $x, y \in \overline{\mathbb{F}}_q$

$$\phi_q(x+y) = \phi_q(x) + \phi_q(y) \quad \text{e} \quad \phi_q(xy) = \phi_q(x)\phi_q(y);$$

3. *se* $a \in \overline{\mathbb{F}}_q$*, allora*

$$a \in \mathbb{F}_{q^m} \quad \Longleftrightarrow \quad \phi_q^m(a) = a.$$

Come si vedrà nella sezione 3.5, l'isomorfismo di Frobenius risulta molto utile nel calcolo della cardinalità dell'insieme dei punti del gruppo su una curva ellittica. Abbiamo tuttavia deciso di introdurre adesso l'isomorfismo di Frobenius perché importante nell'attacco MOV (vedi sezione 3.3.4) al logaritmo discreto.

3.3 Il problema del logaritmo discreto

Sia G un gruppo finito e siano $\mathbf{a}, \mathbf{b} \in G$. Supponiamo di sapere che

$$k\mathbf{a} = \mathbf{b}$$

per qualche intero k. Il problema consiste nel riuscire a trovare tale intero conoscendo solamente gli elementi **a** e **b**. Nel caso particolare in cui G sia una curva ellittica, gli elementi in questione sono due punti della curva.

Un metodo per attaccare il logaritmo discreto è quello di provare tutti i valori di k fino ad arrivare a quello giusto: chiaramente questo metodo diventa infattibile per valori di k troppo alti. Abbiamo quindi bisogno di utilizzare tecniche particolari, alcune studiate ad hoc per le curve ellittiche, altre valide in generale. Inizieremo la trattazione con il secondo gruppo (Pohlig-Hellman, Baby step–Giant step, Pollard), per poi passare al primo (MOV e curve anomale).

3.3.1 Il metodo di Pohlig ed Hellman

Prima di procedere con la descrizione del metodo enunciamo il teorema cinese del resto.

Teorema 3.10. *Siano* n, m *due interi positivi con* $\text{MCD}(m, n) = 1$*. Allora l'applicazione*

$$f : \mathbb{Z}_{mn} \to \mathbb{Z}_n \times \mathbb{Z}_m \quad \textit{data da} \quad f(a \bmod mn) = (a \bmod n, a \bmod m)$$

è un isomorfismo.

Corollario 3.11. *Siano* n, m *due interi positivi con* $\text{MCD}(m, n) = 1$ *e siano* $\alpha, \beta \in \mathbb{Z}$*. Allora esiste* $z \in \mathbb{Z}$ *tale che*

$$z \equiv \alpha \bmod n \quad e \quad z \equiv \beta \bmod m.$$

Inoltre l'intero z *è unico modulo* nm*.*

Sia G un gruppo finito e sia n la sua cardinalità. Possiamo scomporre n in fattori primi ed ottenere

$$n = \prod_{i=1}^{t} q_i^{e_i}$$

L'idea di Pohlig ed Hellman è quella di trovare $k \bmod q_i^{e_i}$, utilizzando poi il teorema cinese del resto per calcolare $k \bmod n$.

Per semplicità, fissato i, siano $q = q_i$ ed $e = e_i$. Scriviamo $k = k_0 q^0 + k_1 q^1 + \cdots + k_e q^{e-1}$ con $0 \leqslant k_r < q$. Siano inoltre $\mathbf{a}$ e $\mathbf{b}$ i punti della curva tali che $k\mathbf{a} = \mathbf{b}$. Possiamo implementare il metodo in questo modo:

- calcoliamo $T = \left\{ j \left(\frac{n}{q}\mathbf{a} \right) \text{ con } 0 \leqslant j < q \right\}$;
- calcoliamo $\frac{n}{q}\mathbf{b}$, che sarà l'elemento $k_0 \left(\frac{n}{q}\mathbf{a} \right)$ di T: infatti

$$\begin{aligned} \frac{n}{q}\mathbf{b} &= \frac{n}{q}k\mathbf{a} = \frac{n}{q}(k_0 q^0 + k_1 q^1 + \cdots + k_e q^{e-1})\mathbf{a} = \\ &= k_0 \frac{n}{q}\mathbf{a} + (k_1 q^0 + k_2 q^1 + \cdots + k_e q^{e-2})n\mathbf{a} = k_0 \frac{n}{q}\mathbf{a} \end{aligned} \tag{3.1}$$

  giacché $n\mathbf{a} = \mathcal{O}$;
- calcoliamo $\mathbf{b}_1 = \mathbf{b} - k_0\mathbf{a}$;
- calcoliamo $\frac{n}{q^2}\mathbf{b}_1$, che sarà l'elemento $k_1 \left(\frac{n}{q}\mathbf{a} \right)$ di T;
- ...;
- calcoliamo $\mathbf{b}_r = \mathbf{b}_{r-1} - k_{r-1} q^{r-1}\mathbf{a}$.
- calcoliamo $\frac{n}{q^{r+1}}\mathbf{b}_r$, che sarà l'elemento $k_r \left(\frac{n}{q}\mathbf{a} \right)$ di T;
- ...;
- il metodo prosegue finché $r = e - 1$.

A questo punto si può calcolare $k = k_0 q^0 + k_1 q^1 + \cdots + k_e q^{e-1}$ modulo $q_i^{e_i}$ e successivamente $k \bmod n$ attraverso il teorema cinese del resto. Il problema viene quindi "spostato" sui sottogruppi di G con un numero primo di elementi. Per questa ragione, se un sistema crittografico si basa sul logaritmo discreto (come ad esempio il sistema di ElGamal), bisogna accertarsi che la cardinalità del gruppo di partenza abbia un divisore primo molto grande.

3.3.2 Il metodo Baby step–Giant step

Descriviamo ora un metodo sviluppato da Shanks per risolvere il problema del logaritmo discreto in un gruppo generico G con n elementi. Data la semplificazione di Pohling-Hellman, possiamo supporre n primo, benché questa informazione non sia richiesta dall'algoritmo.

Siano **a**, **b**, k e n come nella sezione precedente. Indicheremo con $\lceil x \rceil$ il più piccolo intero maggiore o uguale a x. È noto dall'aritmetica che possiamo scrivere

$$k = \lceil \sqrt{n} \rceil c + d$$

con $0 \leqslant c, d < \lceil \sqrt{n} \rceil$. Il concetto che sta alla base del metodo è quello di provare i valori di k passando in rassegna i valori di c e d. L'algoritmo termina dopo $\lceil \sqrt{n} \rceil$ operazioni ed ha bisogno di memorizzare $\lceil \sqrt{n} \rceil$ valori; non può quindi essere utilizzato per valori di n troppo grandi.

Si procede come segue:

- fissiamo $m \geqslant \lceil \sqrt{n} \rceil$ e calcoliamo $m\mathbf{a}$;
- calcoliamo e memorizziamo la lista degli $i\mathbf{a}$ per $0 \leqslant i < m$;
- calcoliamo i punti $\mathbf{b} - jm\mathbf{a}$ per $j = 0, 1, \ldots, m-1$ finché non ne troviamo uno nella lista precedentemente calcolata.

Se $i\mathbf{a} = \mathbf{b} - jm\mathbf{a}$ abbiamo $\mathbf{b} = k\mathbf{a}$ con $k = i + jm \bmod n$.

Il nome deriva dal fatto che il punto $i\mathbf{a}$ viene calcolato aggiungendo $\mathbf{a}$ al valore $(i-1)\mathbf{a}$ (Baby step, ovvero passo minuscolo), mentre il punto $\mathbf{b}-jm\mathbf{a}$ viene calcolato aggiungendo $-m\mathbf{a}$ al valore $\mathbf{b}-(j-1)m\mathbf{a}$ (Giant step, ovvero passo gigantesco).

Perché il metodo funziona? Dobbiamo dimostrare che attraverso i "passi giganteschi" arriviamo sicuramente ad incontrare un punto dei "passi minuscoli". Ma, come detto precedentemente, noi possiamo scrivere $k = ma + b$ tramite un'unica coppia a, b che soddisfi $0 \leqslant a, b < m$. Quindi quando $i = b$ e $j = a$ la condizione $k = i + jm$ è sicuramente verificata.

Il grande svantaggio di questo metodo è la memorizzazione di almeno $\lceil \sqrt{n} \rceil$ valori. Tuttavia può essere utile insieme con la scomposizione di Pohlig-Hellman, che divide il problema in problemi più piccoli. È da notare infine che per applicare questo metodo non abbiamo bisogno di sapere precisamente il valore di n, ma solamente un maggiorante di esso. Ad esempio il teorema 3.15 di Hasse ci dice che una qualsiasi curva $E/\mathbb{F}_q$ ha al massimo $q + 1 + 2\sqrt{q}$ punti, quindi possiamo usare quel valore come maggiorante.

Si prenda ad esempio $G = E/\mathbb{F}_{41}$ con $E : y^2 = x^3 + 2x + 1$. Siano $P = (0,1)$ e $Q = (30,40)$. Per il teorema di Hasse la curva ha al massimo 54 punti, quindi possiamo prendere $m = 8$. I punti iP sono:

$$(0,1), (1,39), (8,23), (38,38), (23,23), (20,28), (26,9).$$

Calcoliamo $\mathbf{b} - jm\mathbf{a}$ per $j = 0, 1, 2$ e otteniamo

$$(30,40), (9,25), (26,9)$$

sul quale ci fermiamo: abbiamo infatti trovato un punto nell'elenco dei "passi minuscoli", precisamente $7P$. Poiché $j = 2$ soddisfa alla proprietà, troviamo

$$(30,40) = (7 + 2 \cdot 8)P = 23P$$

da cui deduciamo $k = 23$.

3.3.3 Il metodo ρ di Pollard

Siano $\mathbf{a}$, $\mathbf{b}$, k e n come nelle sezioni precedenti. Per applicare il metodo ρ di Pollard bisogna innanzi tutto scegliere una funzione "casuale" $f : G \to G$. Vedremo in seguito come sceglierla. Inoltre bisogna scegliere un elemento $\mathbf{a}_0 \in E$ di partenza e calcolare la successione $\mathbf{a}_i = f(\mathbf{a}_{i-1})$. Poiché il gruppo è finito, esisteranno i_0 e j_0 (con $i_0 < j_0$) tali che $\mathbf{a}_{i_0} = \mathbf{a}_{j_0}$. Quindi

$$\mathbf{a}_{j_0+\ell} = \mathbf{a}_{i_0+\ell}$$

per ogni $\ell \geqslant 0$. La successione $\mathbf{a}_i$ è chiaramente periodica di periodo $j_0 - i_0$. Se la funzione è scelta in modo abbastanza casuale, la coppia (i_0, j_0) verrà trovata dopo una media di $\sqrt{n}$ iterazioni.

Un'implementazione del sistema spiegato finora prevede il salvataggio di tutti gli $\mathbf{a}_i$ precedenti ad $\mathbf{a}_{i_0}$, che sono in numero di $\sqrt{n}$ (circa, visto che si tratta di un metodo stocastico), quindi apparentemente non abbiamo vantaggi rispetto al metodo Baby step–Giant step. In realtà possiamo risparmiare spazio memorizzando solamente alcuni elementi $\mathbf{a}_i$ (detti *privilegiati*), ad esempio quelli per cui la rappresentazione binaria della coordinata x termini con t zeri. In questo modo salviamo solamente un numero ogni 2^t. Se accadesse che $\mathbf{a}_i = \mathbf{a}_j$ ma $\mathbf{a}_i$ non fosse privilegiato, ci sarebbe un numero privilegiato $\mathbf{a}_{i+\ell}$ con $1 \leqslant \ell \leqslant 2^t$. Quindi $\mathbf{a}_{j+\ell} = \mathbf{a}_{i+\ell}$; abbiamo in pratica trovato una corrispondenza con un numero privilegiato dopo solo qualche calcolo in più, fatto tuttavia compensato da un minor numero di dati immagazzinati.

Come si può prendere la funzione f? Il metodo funziona tanto meglio tanto più la f è casuale, quindi bisogna trovare una candidata particolarmente adatta. Un metodo è quello di dividere il gruppo G in s sottoinsiemi disgiunti $S_1, S_2, \ldots, S_s$ più o meno della stessa dimensione (di solito $s = 20$ e $|S_i| \simeq n/20$) e scegliere $2s$ interi casuali a_i, b_i. Sia poi

$$\mathbf{m}_i = a_i\mathbf{a} + b_i\mathbf{b}.$$

Possiamo ora definire la funzione "casuale" f come

$$f(\mathbf{g}) = \mathbf{g} + \mathbf{m}_i \quad \text{se } \mathbf{g} \in S_i.$$

Infine si scelgono due interi casuali a_0, b_0 e si sceglie il punto iniziale $\mathbf{a}_0 = a_0\mathbf{a} + b_0\mathbf{b}$. Ogni elemento $\mathbf{a}_i$ della successione può essere scritto come combinazione di $\mathbf{a}$ e $\mathbf{b}$, in quanto godono della stessa proprietà anche $\mathbf{a}_0$ ed $\mathbf{m}_i$ per ogni i.

Come facciamo a trovare k? Se troviamo una corrispondenza $\mathbf{a}_{i_0} = \mathbf{a}_{j_0}$, allora abbiamo:

$$\begin{aligned} \mathbf{a}_{j_0} &= \mathbf{a}_{i_0} \\ u_{j_0}\mathbf{a} + v_{j_0}\mathbf{b} &= u_{i_0}\mathbf{a} + v_{i_0}\mathbf{b} \\ (v_{j_0} - v_{i_0})\mathbf{b} &= (u_{i_0} - u_{j_0})\mathbf{a}. \end{aligned}$$

Se $d = \text{MCD}(v_{j_0} - v_{i_0}, n)$, essendo $k\mathbf{a} = \mathbf{b}$, abbiamo

$$k \equiv (v_{j_0} - v_{i_0})^{-1}(u_{i_0} - u_{j_0}) \bmod n/d.$$

In questo modo otteniamo d possibili valori di k. Poiché solitamente d è relativamente piccolo, possiamo trovare k esaminandone tutti i d possibili valori. Nella crittografia molto spesso n è primo, il che vuol dire che $d = 1$ oppure $d = n$: nel primo caso la soluzione è unica, mentre nel secondo caso otteniamo una relazione banale, quindi siamo costretti a ripetere il procedimento con valori diversi dei coefficienti a_i e b_i.

Un'ultima osservazione riguarda la definizione di f. Non abbiamo bisogno di fornire un valore per $f(\mathcal{O})$: se ci trovassimo infatti in una situazione in cui $a\mathbf{a} + b\mathbf{b} = \mathcal{O}$ possiamo analogamente trovare k, purché ovviamente non incappiamo in casi in cui $\text{MCD}(b, n)$ sia troppo alto.

3.3.4 Il metodo Menezes, Okamoto, Vanstone

L'attacco MOV utilizza l'accoppiamento di Weil per ridurre il problema del logaritmo discreto nelle curve ellittiche ad un problema di logaritmo discreto nel rispettivo gruppo moltiplicativo. Questo praticamente "vanifica" l'utilizzo delle curve ellittiche in quel particolare caso. Queste ultime, infatti, ci permettono di utilizzare gruppi finiti con un minor numero di elementi, in quanto in generale più difficili da trattare; se però il problema viene ridotto a quello sul corrispettivo gruppo numerico, i metodi appena descritti hanno più facilmente successo.

Sia $E/\mathbb{F}_q$ una curva ellittica e siano $P = \mathbf{a}$ e $Q = \mathbf{b}$ due punti di E. Sia n l'ordine di P. Assumiamo inoltre che $\text{MCD}(n, q) = 1$. Il nostro scopo è trovare un intero k tale che $Q = kP$.

Per prima cosa bisogna assicurarsi che k esista.

Proposizione 3.12. *Sotto le ipotesi appena fatte,* k *esiste se e solo se* $nQ = \mathcal{O}$ *ed* $e_n(P, Q) = 1$*, dove* e_n *è l'accoppiamento di Frobenius (introdotto nel teorema 3.5).*

Si scelga ora m tale che

$$E[n] \subseteq E/\mathbb{F}_{q^m}$$

Poiché tutti i punti di $E[n]$ hanno coordinate in $\overline{\mathbb{F}}_q = \bigcup_{j>0} \mathbb{F}_{q^j}$, tale m deve esistere. Dal corollario 3.7 il gruppo μ_n è contenuto in $\mathbb{F}_{q^m}$. Tutti i calcoli che andiamo a presentare saranno in $\mathbb{F}_{q^m}$. L'algoritmo è il seguente:

- si scelga un punto casuale $R \in E/\mathbb{F}_{q^m}$;
- si calcoli l'ordine m di R;
- sia $d = \text{MCD}(n, m)$ e sia $R_1 = \frac{m}{d}R$. R_1 ha quindi ordine d, che divide n, quindi $R_1 \in E[n]$;
- si calcolino $\zeta_1 = e_n(P, R_1)$ e $\zeta_2 = e_n(Q, R_1)$. Entrambi ζ_1 e ζ_2 sono in μ_d;
- si risolva il logaritmo discreto $\zeta_2 = \zeta_1^k$ in $\mathbb{F}_{q^m}$. Questo fornisce $k \bmod d$;

- si ripeta il procedimento con scelte di R diverse finché il minimo comune multiplo dei vari d ottenuti sia n; questo determinerà univocamente $k \bmod n$.

Il trucco è quindi quello di spostare il problema del logaritmo discreto delle curve ellittiche sul logaritmo discreto in campi finiti. L'unico inconveniente è il valore m: se è troppo alto, il problema si complica invece di semplificarsi. Tuttavia per le curve supersingolari possiamo prendere m molto piccoli: se q è un numero primo $p \geqslant 5$, allora $m = 2$, altrimenti $m = 3, 4$ o 6; vale in particolare la proposizione:

Proposizione 3.13. *Sia $E/\mathbb{F}_q$ una curva ellittica e sia $q + 1$ la sua cardinalità. Sia inoltre n un numero intero arbitrario. Se esiste un punto $P \in E/\mathbb{F}_q$ di ordine n, allora $E[n] \subseteq E/\mathbb{F}_{q^2}$.*

3.3.5 Le curve anomale

Oltre che per le curve supersingolari, anche per un'altra categoria di esse il problema del logaritmo discreto può essere ridotto al logaritmo discreto sui campi di Galois. Queste curve (dette **anomale**) sono caratterizzate dal fatto che la loro cardinalità è uguale alla cardinalità del campo su cui sono costruite. In altre parole $|E/\mathbb{F}_q| = q$. Non ci addentreremo nella descrizione di questa categoria.

3.4 Scegliere la curva

La prima cosa da fare per poter procedere all'operazione di crittazione è scegliere una curva adatta allo scopo. Vediamo come compiere questa operazione senza incappare nei punti di forza degli attacchi appena descritti.

Un algoritmo per selezionare una buona curva può essere riassunto come segue:

1. scegliere un intero piccolo $\overline{s}$ e un campo finito $\mathbb{F}_q$;
2. scrivere l'equazione di una curva E con coefficienti aleatori in $\mathbb{F}_q$;
3. calcolare $|E/\mathbb{F}_q|$;
4. controllare la supersingolarità, in caso positivo tornare al punto 2;
5. controllare che la curva non sia anomala, in caso contrario tornare al punto 2;
6. eseguire il test di fattorizzazione con l'intero $\overline{s}$; se non si riesce tornare al punto 2;
7. sia $|E/\mathbb{F}_q| = sr$ con r il primo più grande che divide $|E/\mathbb{F}_q|$; se $s > \overline{s}$ tornare al punto 2;
8. E è la curva cercata.

L'intero $\overline{s}$ serve per stabilire quando il test di fattorizzazione del punto 6 fallisce o meno. Si procede in questo modo: si prova a dividere $|E/\mathbb{F}_q|$ per tutti i primi più piccoli di $\overline{s}$ finché si ottiene un numero r, divisore di $|E/\mathbb{F}_q|$, che non è più diviso da nessun primo minore di $\overline{s}$; si esegue quindi un test di primalità su r; se il test fallisce si torna al punto 2. Il calcolo della cardinalità della curva sarà argomento della prossima sezione.

3.5 Contare i punti

La semplificazione di Pohlig-Hellman per il calcolo del logaritmo discreto funziona tanto meglio quanto la cardinalità del gruppo è scomponibile in fattori primi piccoli. Quello che ci serve è dunque conoscere il numero di punti del gruppo utilizzato.

Definizione 3.14. *Sia data* $E/\mathbb{F}_q$ *curva ellittica e sia* $|E/\mathbb{F}_q| = q+1-t$ *il numero dei suoi punti (cardinalità). L'intero* t *è detto* ***traccia di Frobenius****.*

Un primo risultato fondamentale è il teorema di Hasse, che pone un intervallo in cui cade la cardinalità del gruppo.

Teorema 3.15. *Sia* E/K *una curva ellittica. La traccia di Frobenius* t *soddisfa la disequazione*

$$|t| \leqslant 2\sqrt{q}.$$

Sia E una curva ellittica definita su un campo finito $\mathbb{F}_q$. L'automorfismo di Frobenius (vedere sezione 3.2) agisce sui punti di $E/\overline{\mathbb{F}}_q$ in questo modo:

$$\phi_q(x,y) = (x^q, y^q) \quad \text{e} \quad \phi_q(\mathcal{O}) = \mathcal{O}.$$

Proposizione 3.16. *Sia* E *una curva ellittica definita su* $\mathbb{F}_q$*, e sia* (x,y) *un punto di* $E/\overline{\mathbb{F}}_q$*. Allora valgono:*

1. $\phi_q(x,y) \in E/\overline{\mathbb{F}}_q$ *cioè* ϕ_q *si restringe ad una mappa da* $E/\overline{\mathbb{F}}_q$ *a* $E/\overline{\mathbb{F}}_q$*;*
2. $\phi_q : E/\overline{\mathbb{F}}_q \to E/\overline{\mathbb{F}}_q$ *è un omomorfismo di gruppi e una mappa razionale, cioè è un* ***endomorfismo di*** $E/\overline{\mathbb{F}}_q$*;*
3. $(x,y) \in E/\mathbb{F}_q$ *se e solo se* $\phi_q(x,y) = (x,y)$.

Teorema 3.17. *Sia* $E/\mathbb{F}_q$ *una curva ellittica. Allora* t *è la traccia di Frobenius di* E *se e solo se vale*

$$\phi_q^2 - t\phi_q + q = 0 \tag{3.2}$$

come endomorfismo di $E/\overline{\mathbb{F}}_q$*, dove* ϕ_q *è mappa di Frobenius. Inoltre* t *è l'unico intero che rende vera la (3.2).*

In altre parole, se $(x,y) \in E/\overline{\mathbb{F}}_q$*, allora*

$$\left(x^{q^2}, y^{q^2}\right) - t(x^q, y^q) + q(x,y) = \mathcal{O}$$

e t *è l'unico intero che rende vera questa equazione per ogni punto* (x,y) *della curva.*

Il teorema 3.15 fornisce un intervallo entro il quale cade il numero di punti di una generica curva ellittica su un campo $\mathbb{F}_q$. Tuttavia se $q = p^m$ con p sufficientemente piccolo risulta particolarmente pratico un teorema che calcola il valore di t per $\mathbb{F}_q$ dato il valore t per $\mathbb{F}_p$.

Teorema 3.18. *Sia* $E/\mathbb{F}_p$ *una curva ellittica e sia* t_p *la sua traccia di Frobenius. Si risolva l'equazione di secondo grado data da* $x^2 - t_p x + p = (x-\alpha)(x-\beta)$. *Allora*

$$t_{p^m} = \alpha^m + \beta^m$$

dove t_{p^m} *è la traccia di Frobenius di* $E/\mathbb{F}_{p^m}$.

Dimostrazione. Innanzi tutto facciamo vedere che la quantità $\alpha^n + \beta^n$ è intera. Per fare ciò si consideri la funzione $s_n = \alpha^n + \beta^n$. Abbiamo $s_0 = 2$ e $s_1 = \alpha + \beta = a$. Poiché α e β sono soluzioni, $\alpha^2 - t_p\alpha + p = 0$, da cui, moltiplicando entrambi i membri per α^{n-1} otteniamo $\alpha^{n+1} = t_p\alpha^n - p\alpha^{n-1}$. Analogamente si procede per β. Sommando le due relazioni possiamo concludere che

$$s_{n+1} = \alpha^{n+1} + \beta^{n+1} = t_p\alpha^n - p\alpha^{n-1} + t_p\beta^n - p\beta^{n-1} = t_p s_n - p s_{n-1}. \tag{3.3}$$

Applicando un ragionamento induttivo su s_n possiamo concludere che esso è sempre un numero intero.

Sia ora

$$f(x) = (x^n - \alpha^n)(x^n - \beta^n) = x^{2n} - (\alpha^n + \beta^n)x^n + p^n.$$

Il polinomio $x^2 - t_p x + p = (x-\alpha)(x-\beta)$ chiaramente divide $f(x)$. Il quoziente $Q(x)$ sarà un polinomio con coefficienti interi, in quanto il coefficiente di grado più elevato di $x^2 - t_p x + p$ è 1 e lo stesso polinomio e $f(x)$ hanno coefficienti interi. Quindi, applicando la f alla mappa di Frobenius ϕ_p, otteniamo

$$f(\phi_p) = \phi_p^{2n} - (\alpha^n + \beta^n)\phi_p^n + p^n = Q(\phi_p)(\phi_p^2 - t_p\phi_q + p) = 0 \tag{3.4}$$

per il teorema 3.17. Dalla relazione $\phi_p^n = \phi_{p^n}$ e sempre dal teorema 3.17 segue che esiste un unico intero k tale che

$$\phi_p^{2n} - k\phi_p^n + p^n = \phi_{p^n}^2 - k\phi_{p^n} + p^n = 0.$$

La (3.4) implica che $k = \alpha^n + \beta^n$ e il teorema 3.17 ci dice che $k = t_{p^m}$, da cui la tesi. □

Un esempio chiarirà l'importanza di questo teorema.

Sia quindi E la curva $f(x,y) = y^2 + xy + x^3 + 1 = 0$. Calcoliamo ora quanti punti possiede su $\mathbb{F}_2$.

$$\begin{aligned} & \mathcal{O} \in E/\mathbb{F}_2 \\ x = 0 \implies & y^2 + 1 = 0, \text{ quindi } y = 1 \\ x = 1 \implies & y^2 + y = 0, \text{ quindi } y = 0, 1 \end{aligned}$$

Possiamo concludere che $E/\mathbb{F}_2 = \{\mathcal{O}, (0,1), (1,0), (1,1)\}$. La traccia di Frobenius della curva su $\mathbb{F}_2$ è $t_2 = -1$. Applichiamo ora il teorema 3.18 con $n = 2$ per ottenere i punti della curva $E/\mathbb{F}_4 = E/\mathbb{F}_{2^2}$. Partendo da $t_2 = -1$ otteniamo il polinomio

$$x^2 + x + 2 = \left(x - \frac{-1+\sqrt{-7}}{2}\right)\left(x - \frac{-1-\sqrt{-7}}{2}\right)$$

che ci servirà per calcolare t_4. Applicando la (3.3), possiamo facilmente calcolare s_2, ovvero $\alpha^2 + \beta^2$.

$$s_2 = t_2 s_1 - 2s_0 = 1 - 4 = -3$$

e quindi $|E/\mathbb{F}_4| = 1 + 4 - (-3) = 8$. Verifichiamo ora il risultato appena ottenuto contando manualmente i punti di $E/\mathbb{F}_4$. Ricordiamo che il campo $\mathbb{F}_4$ è fatto in questo modo:

$+$	0	1	c_1	c_2
0	0	1	c_1	c_2
1	1	0	c_2	c_1
c_1	c_1	c_2	0	1
c_2	c_2	c_1	1	0

$\times$	0	1	c_1	c_2
0	0	0	0	0
1	0	1	c_1	c_2
c_1	0	c_1	c_2	1
c_2	0	c_2	1	c_1

Ora vediamo come si comporta la curva E su questo campo.

$$\begin{aligned} & \mathcal{O} \in E/\mathbb{F}_2 \\ x = 0 \quad &\Longrightarrow \quad y^2 + 1 = 0, \text{ quindi } y = 1 \\ x = 1 \quad &\Longrightarrow \quad y^2 + y = 0, \text{ quindi } y = 0, 1 \\ x = c_1 \quad &\Longrightarrow \quad y^2 + c_1 y = 0, \text{ quindi } y = 0, c_1 \\ x = c_2 \quad &\Longrightarrow \quad y^2 + c_2 y = 0, \text{ quindi } y = 0, c_2 \end{aligned}$$

Possiamo elencare i punti, $E/\mathbb{F}_4 = \{\mathcal{O}, (0,1), (1,0), (1,1), (c_1,0), (c_1,c_1), (c_2,0), (c_2,c_2)\}$, in tutto 8, come previsto dal teorema 3.18.

Diventa quindi molto semplice calcolare il numero di punti di una curva su un campo $\mathbb{F}_{p^m}$ con p piccolo, anche se m è molto alto.

Nel caso invece in cui p sia molto elevato il problema è più complesso e il calcolo (peraltro non sempre possibile) richiede strumenti che esulano dagli obiettivi di questo testo.

3.6 Utilizzo attuale delle curve ellittiche

È utile sapere che nella pratica i campi utilizzati sono solamente $\mathbb{F}_p$ con p primo (molto grande) e $\mathbb{F}_{2^m}$, con m elevato. I primi sono comodi per l'aritmetica relativamente semplice; i secondi, invece, ammettendo una rappresentazione binaria per gli elementi, rendono più semplice il lavoro da parte di un calcolatore. I campi del tipo $\mathbb{F}_{p^m}$ sono ugualmente comodi dal punto di vista della sicurezza, ma non hanno nessun vantaggio particolare rispetto a quelli descritti sopra, quindi non vengono presi in considerazione.

Nella pratica le curve ellittiche sono già utilizzate ad esempio nella rete Berlino-Bonn per lo scambio di documenti riservati tra i due capoluoghi tedeschi. L'ambito dove probabilmente le curve ellittiche avranno più successo è quello delle *smartcard*, tessere simili alle carte magnetiche, con in aggiunta un

piccolissimo processore in grado di eseguire semplici calcoli. Essendo infatti i gruppi generati dalle curve ellittiche più piccoli rispetto a quelli numerici tradizionali, è possibile inserire questa tecnologia anche in apparati di dimensioni microscopiche. In Austria, ad esempio, è previsto per la fine dell'anno il lancio di una *smartcard*, la cui sicurezza si basa sulle curve ellittiche, utilizzabile in banca come firma digitale.

Un'altro aspetto interessante delle curve ellittiche è il cosiddetto "coefficiente di invecchiamento". In un sistema crittografico moderno la sicurezza dipende dalla lunghezza della chiave; il coefficiente di invecchiamento misura la variazione della lunghezza della chiave nel tempo mantenendo la sicurezza costante. Se assumiamo che la potenza di calcolo di un computer cresca secondo la legge empirica di Moore (cioè raddoppia ogni 18 mesi), allora i sistemi basati sulle curve ellittiche rimarrebbero sicuri con una chiave di 300 bit fin dopo il 2040, quando il sistema RSA avrebbe bisogno di una chiave di 3000 bit, dieci volte superiore. Anche allo stato attuale, comunque, a parità di sicurezza un cifrario RSA necessita di almeno 900 bit di chiave, mentre sono sufficienti 150-200 bit per un sistema che usi le curve ellittiche.

Bibliografia

[1] Gordon B. Agnew, Ronald C. Mullin e Scott A. Vanstone. An implementation of elliptic curve cryptosystems over $\mathbb{F}_{2^{155}}$, *IEEE Journal on selected areas in communications* **11** (1993), no. 5, 804–813.

[2] Carlo Bertoluzza. *Dispense del corso di Crittografia*, inedito (2003).

[3] Thomas Beth e Frank Schaefer. Non supersingular elliptic curves for public key cryptosystems, *Advances in Cryptology—EUROCRYPT '91 (Brighton, 1991)*, 316–327, Lecture Notes in Comput. Sci., **547**, Springer, 1992.

[4] Ian Blake, Gadiel Seroussi e Nigel Smart. *Elliptic curves in cryptography*, Cambridge University Press, 1999.

[5] Matthias Büger, Bernhard Esslinger, Bartol Filipovics e Roger Oyono. *Cryptool script: mathematics and cryptography–chapter 6*, inedito (2003).

[6] Egbert Brieskorn e Horst Knörrer. *Plane algebraic curves*, Birkhäuser, 1986.

[7] Whitfield Diffie e Martin E. Hellman. New directions in cryptography, *IEEE Trans. Inform. Theory* **IT-22** (1976), no. 6, 644–654.

[8] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Inform. Theory* **31** (1985), no. 4, 469–472.

[9] Taher ElGamal. A subexponential-time algorithm for computing discrete logarithms over GF(p^2), *IEEE Trans. Inform. Theory* **31** (1985), no. 4, 473–481.

[10] Neal Koblitz. *A course in number theory and cryptography*, Springer-Verlag, 1987.

[11] Neal Koblitz. Constructing elliptic curve cryptosystems in characteristic 2, *Advances in Cryptology—CRYPTO '90 (Santa Barbara, CA, 1990)*, 156–167, Lecture Notes in Comput. Sci., **537**, Springer, 1991.

[12] Neal Koblitz. Elliptic curve cryptosystems, *Math. Comp.* **48** (1987), no. 177, 203–209.

[13] Arjen K. Lenstra e Eric R. Verheul. Selecting cryptographic key sizes, *J. Cryptology* **14** (2001), no. 4, 255–293.

[14] Rudolf Lidl e Harald Niederreiter. *Introduction to finte fields and their applications*, Cambridge University Press, 1986.

[15] Alfred J. Menezes. *Elliptic curve public key cryptosystems*, Kluwer Academic Publishers, 1993.

[16] Alfred J. Menezes, Tatsuaki Okamoto e Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Trans. Inform. Theory* **39** (1993), no. 5, 1639–1646.

[17] Alfred J. Menezes e Scott A. Vanstone. Isomorphism classes of elliptic curves over finite fields of characteristic 2, *Utilitas Math.* **38** (1990), 135–153.

[18] Alfred J. Menezes, Scott A. Vanstone e Robert J. Zuccherato. Counting points on elliptic curves over $\mathbb{F}_{2^m}$, *Math. Comp.* **60** (1993), no. 201, 407–420.

[19] Ralph C. Merkle. Secure communications over insecure channels, *CACM* **21** (1978), no. 4, 294–299.

[20] Victor S. Miller. Use of elliptic curves in cryptography, *Advances in Cryptology—CRYPTO '85 (Santa Barbara, CA, 1985)*, 417–426, Lecture Notes in Comput. Sci., **218**, Springer, 1986.

[21] John M. Pollard. Monte Carlo methods for index computation $\bmod p$, *Math. Comp.* **32** (1978), no. 143, 918–924.

[22] Hans-Georg Rück. A note on elliptic curves over finite fields, *Math. Comp.* **49** (1987), no. 179, 301–304.

[23] Edoardo Sernesi. *Geometria 1*, Bollati Boringhieri, 2000.

[24] René Schoof e Lambertus van Geemen. *Note per il corso di algebra*, inedito (2001).

[25] Joseph H. Silverman. *The arithmetic of elliptic curves*, Springer-Verlag, 1986.

[26] Daniel Shanks. Class number, a theory of factorization, and genera. *1969 Number Theory Institute (Stony Brook, NY, 1969)*, 415–440. *Amer. Math. Soc., Providence, RI*, 1971.

[27] Nigel P. Smart. The discrete logarithm problem on elliptic curves of trace one, *J. Cryptology* **12** (1999), no. 3, 193–196.

[28] Douglas R. Stinson, *Cryptography: theory and practice*. Chapman & Hall/CRC, 2002.

[29] Lawrence C. Washington. *Elliptic curves: number theory and cryptography*, Chapman & Hall/CRC, 2003.

[30] William C. Waterhouse. Abelian varieties over finite fields, *Ann. Sci. École Norm. Sup.* **2** (1969), no. 4, 521–560.