UNIVERSITÀ DEGLI STUDI "ROMA TRE"
Dipartimento di Matematica e Fisica
Corso di Laurea Magistrale in Matematica

Sintesi della Tesi di Laurea Magistrale

Curve Ellittiche in Crittografia

Candidata:
Luciana Longo

Relatore:
Prof. Francesco Pappalardi

Ottobre 2017
Anno Accademico 2016/2017

Sintesi

La crittografia è l'arte di nascondere i messaggi, in modo tale che, anche se intercettati da terzi, risultino illeggibili. Risultati interessanti si sono ottenuti basando i crittosistemi tradizionali su curve ellittiche.
Nei tre capitoli di cui ora si darà solo una sintesi, si è posta l'attenzione proprio sullo studio delle curve ellittiche e sulla loro applicazione alla crittografia a chiave pubblica.

Il modello generale della Crittografia prevede un mittente "Alice", un destinatario "Bob" ed un probabile attaccante "Eve". L'idea è la seguente: Il mittente vuole inviare un messaggio al destinatario in modo segreto, in maniera tale che il nemico non sia in grado di conoscere il messaggio. Il nemico nota la trasmissione del messaggio criptato, ma non ha modo di conoscere il messaggio originale, poichè il mittente trasforma il "testo in chiaro" M, in un testo cifrato C, e solo il destinatario deve essere in grado di decifrarlo, ovvero di ottenere M conoscendo C. Da qui nasce l'esigenza di utilizzare una chiave segreta, nota solo al destinatario.

In dettaglio, la tesi è così organizzata.
Nel **primo capitolo**, si richiameranno i concetti base della crittografia a chiave simmetrica e a chiave pubblica. A partire dai crittosistemi tradizionali, come RSA e il crittosistema a chiave pubblica ElGamal, si arriverà a descrivere gli analoghi algoritmi utilizzando le curve ellittiche per scoprirne il vantaggio.

La sicurezza dei crittosistemi è basata sui seguenti problemi:

1. **Problema della fattorizzazione degli interi (IFP)**: Dato un numero intero $n = pq$, con p e q primi molto grandi, determinare p e q. Su questo problema si basa il crittosistema RSA;

2. **Problema del logaritmo discreto (DLP)**: Calcolare il logaritmo x di un elemento $b = a^x$ su campi finiti. Su tale problema si basa, ad esempio, il crittosistema di ElGamal;

3. **Problema del logaritmo discreto su curve ellittiche (ECDLP)**: Questo problema è una forma generalizzata del DLP sui punti di una curva ellittica. Su questo problema si basa l'analogo crittosistema di ElGamal su curve ellittiche.

Una delle applicazioni più importanti della crittografia a chiave pubblica è la cosiddetta *firma digitale* o *elettronica*. La firma digitale consiste di un procedimento matematico che permette di associare un documento elettronico al suo legittimo proprietario e che produce una sequenza binaria, detta appunto firma. Ovviamente, per garantire la veridicità del documento firmato, il processo di firma deve utilizzare un informazione nota solo al proprietario, ovvero la sua chiave privata. Il destinatario, per verificare l'autenticità della firma, utilizzerà la chiave pubblica del mittente. Pertanto, se Alice vuole inviare un messaggio con la propria firma a Bob, detto M tale messaggio, calcola $S = Sign_{Priv(A)}(M)$ ed invia la coppia (M, S); a questo punto, Bob applica l'algoritmo $Ver_{Priv(A)}(M, S)$ per verificare che S sia effettivamente la firma di Alice. Come possiamo notare, l'agoritmo di firma utilizza la chiave privata di Alice, mentre quello di verifica usa quella pubblica. Nella pratica, la firma digitale non viene applicata all'intero documento, bensì ad una sua versione compressa ottenuta con l'impiego di funzioni Hash.

In questo primo capitolo si parlerà anche del problema del logaritmo discreto il quale, come il problema della fattorizzazione di interi, è supposto difficile da risolvere,

cioè non sono noti algoritmi che lo risolvono in tempo polinomiale. Per questo motivo, entrambi i problemi (DLP e IFP) sono stati utilizzati come base teorica per la costruzione di sistemi crittografici.
Ad esempio, uno dei crittosistemi a chiave pubblica basati sul problema del logaritmo discreto è quello di ElGamal, il quale si suddivide in tre fasi:

Generazione delle chiavi

1. Bob sceglie un numero primo p molto grande ed una radice primitiva g (mod p);
2. Seleziona un intero casuale b con $2 \leqslant b \leqslant p-1$ e calcola $g^b \pmod p$;
3. La chiave pubblica di Bob è (p, g, g^b) e la sua chiave privata è b.

Cifratura: Alice vuole inviare un messaggio m a Bob, dove $m \in \{0, 1, \ldots, p-1\}$:

1. Alice riceve la chiave pubblica di Bob (p, g, g^b);
2. Sceglie un intero casuale a, con $0 \leqslant a \leqslant p-1$;
3. Calcola $g^a \pmod p$ e $\lambda = m(g^b)^a \pmod p$;
4. Alice invia a Bob il testo cifrato $c = (g^a, \lambda)$.

Decifratura: Per ottenere il messaggio m da c,

1. Bob usa la sua chiave privata b per calcolare $(g^a)^{-b} \equiv (g^a)^{p-1-b} \pmod p$;
2. Poi decifra il messaggio m calcolando $(g^a)^{-b}\lambda \pmod p$.

La decifratura di tale algoritmo è dovuta al fatto che

$$(g^a)^{-b}\lambda \equiv g^{-ab}m(g^b)^a \equiv mg^{ba-ab} \equiv m \pmod p.$$

Il processo di cifratura è non deterministico in quanto il testo cifrato c dipende dal messaggio m e dal valore casuale a scelto da Alice. Ciò vuol dire che uno stesso messaggio può essere cifrato in diversi modi a seconda del valore a scelto.

L'obiettivo del **secondo capitolo** sarà quello di dare una descrizione matematica delle curve ellittiche, in particolare verrà studiata la sua aritmetica e verranno studiati i concetti di ordine di punto e di curva, fondamentali per comprendere meglio i crittosistemi basati sulle curve ellittiche che saranno affrontati nell'ultimo capitolo di questa tesi.

Definizione 0.1. *Una curva ellittica definita su un campo $\mathbb{K}$, con caratteristica diversa da 2 e da 3, può essere descritta come il grafico di un'equazione, detta* ***equazione di Weierstrass****, della forma:*

$$y^2 = x^3 + Ax + B \tag{1}$$

con $A, B \in \mathbb{K}$, in modo che non sia singolare.

Definizione 0.2. *L'insieme di tutti i punti di una curva ellittica definita su un campo $\mathbb{L} \supseteq \mathbb{K}$ è denotata con $E(\mathbb{L})$. Per definizione, questo insieme contiene sempre il punto ∞ ed è definito come segue:*

$$E(\mathbb{L}) = \{\infty\} \cup \{(x, y) \in L \times L \mid y^2 = x^3 + Ax + B\}.$$

Sulle curve ellittiche è definita una sola operazione, ovvero l'operazione di somma.

Teorema 0.0.1. *Sia E una curva ellittica definita da $y^2 = x^3 + Ax + B$. Siano $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ punti su E con $P_1, P_2 \neq \infty$. Definisco $P_1 + P_2 = P_3 = (x_3, y_3)$ come segue:*

1. *Se $x_1 \neq x_2$, allora*

$$x_3 = m^2 - x_1 - x_2, \qquad y_3 = m(x_1 - x_3) - y_1, \text{ dove } m = \frac{y_2 - y_1}{x_2 - x_1}.$$

2. *Se* $x_1 = x_2$ *ma* $y_1 \neq y_2$, *allora* $P_1 + P_2 = \infty$.

3. *Se* $P_1 = P_2$ *e* $y_1 \neq 0$, *allora*

$$x_3 = m^2 - 2x_1, \qquad y_3 = m(x_1 - x_3) - y_1, \text{ dove } m = \frac{3x_1^2 + A}{2y_1}.$$

4. *Se* $P_1 = P_2$ *e* $y_1 = 0$, *allora* $P_1 + P_2 = \infty$.

5. *Se* $P_2 = \infty$, *allora* $P_1 + \infty = \infty$.

Inoltre, definiamo $P + \infty = P$ *per ogni* $P \in E$.

Quello che possiamo notare da questo Teorema è che l'operazione di somma può essere eseguita più volte. In particolare, se P è un punto della curva ellitica e k un intero positivo, si indica con kP la somma:

$$kP = \overbrace{P + \ldots + P}^{k \quad \text{volte}}.$$

Altri fattori importanti, dal punto di vista crittografico, sono i concetti di ordine della curva $E(\mathbb{F}_q)$ e ordine di un punto. Per quanto riguarda l'ordine di una curva, un teorema fondamentale è il Teorema di Hasse, il quale fornisce un intervallo di valori per la cardinalità di $E(\mathbb{F}_q)$. Questo Teorema, però, non è utile a livello crittografico poichè i crittosistemi basati su curve ellittiche selezionano opportunamente la curva e per tale motivo si richiede la conoscenza esatta del suo ordine. Invece, l'ordine del punto $P \in E(\mathbb{F}_q)$ è il più piccolo intero positivo n tale che $nP = \infty$.

In questo capitolo abbiamo anche affrontato il problema del logaritmo discreto su curve ellittiche (ECDPL, *Elliptic Curve Discrete Logaritm Problem*), in quanto, anche in questo caso, la sicurezza dei crittosistemi si incentra sulla sua intrattabilità.

Problema del logaritmo discreto

Sia E una curva ellittica su un campo finito $\mathbb{F}_q$ e sia P un punto della curva di ordine n e $Q = \langle P \rangle$. Il problema consiste nel trovare l'intero $k \in \{0, 1, \ldots, n-1\}$ tale che

$Q = kP$. Tale intero è chiamato *logaritmo discreto* di Q in base P e si indica con $k = \log_P Q$.

Anche per le curve ellittiche, esistono algoritmi che si utilizzano per la risoluzione di tale problema su curve ellittiche. Questi però non sono molto efficienti, quindi la sicurezza dei crittosistemi si basa anche su questo.

Nel **terzo capitolo**, abbiamo discusso ed esaminato nel dettaglio i vari crittosistemi basati su curve ellittiche.

L'uso delle curve ellittiche in crittografia venne introdotto da Victor Miller nel 1985 e da Neil Koblitz nel 1986. Miller e Koblitz non inventarono nuovi sistemi crittografici, ma furono i primi ad implementare i crittosistemi già esistenti servendosi delle curve ellittiche. Più precisamente, Miller propose l'analogo dello scambio chiavi Diffie-Hellman, mentre Kobliz l'analogo del crittosistema ElGamal; pertanto entrambi utilizzarono il gruppo dei punti di una curva ellittica definita su un campo finito per creare crittosistemi basati sul problema del logaritmo discreto. Il vantaggio è che, utilizzando le curve ellittiche, si riducono notevolmente le dimensioni del gruppo, che si traduce in chiavi crittografiche più piccole e implementazioni più veloci, mantenendo però lo stesso grado di sicurezza.
Nel 1991, Koyama, Maurer, Okomoto e Vanstone proposero l'analogo RSA basato su curve ellittiche definite sull'anello $\mathbb{Z}/n\mathbb{Z}$, con n intero composto. In seguito, però, mostrarono che l'analogo basato su curve ellittiche non introduceva miglioramenti rispetto al crittosistema originale.

Il crittosistema basato su curve ellittiche viene definito da un insieme di parametri, detti *parametri di dominio*, che descrivono la curva $E(\mathbb{F}_q)$. Questi parametri vengono scelti sulla base delle scelte implementative ed in modo tale che il problema

del logaritmo discreto su $E(\mathbb{F}_q)$ resista ai possibili attacchi.

Definizione 0.3. *Definiamo $D = (q, FR, S, A, B, P, n, h)$ l'insieme che racchiude tutti i parametri di dominio di un crittosistema basato su curve ellittiche, dove:*

- *q è l'ordine del campo finito $\mathbb{F}_q$.*
- *FR, field representation, indica il polinomio a coefficienti in $\mathbb{F}_p$ mediante il quale si rappresenta il campo.*
- *S è il "random seed" che viene utilizzato in alcuni algoritmi in cui la curva ellittica viene generata in modo casuale. Per alcuni esempi, si veda [3].*
- *$A, B \in \mathbb{F}_q$ che definiscono l'equazione della curva ellittica $E(\mathbb{F}_q)$ (i.e. $y^2 = x^3 + Ax + B$).*
- *$P = (x_p, y_p) \in E(\mathbb{F}_q)$ ha ordine primo e si chiama punto base.*
- *n è l'ordine del punto P.*
- *$h = \#E(\mathbb{F}_q)/n$, detto cofattore.*

A tali parametri verrà associata una chiave. Come abbiamo già detto in precedenza, i crittosistemi basati su curve ellittiche sono a chiave pubblica. La chiave pubblica corrisponde ad un punto Q scelto casualmente nel gruppo generato dal punto P. La corrispondente chiave privata sarà $d = \log_P Q$. É evidente che il calcolo della chiave privata, a partire da quella pubbica, corrisponde al problema del logaritmo discreto basato su curve ellittiche. Dunque è importante che i parametri di dominio vengano scelti in modo da rendere tale problema intrattabile.

Per determinare la coppia di chiavi (Q,d), utilizziamo il seguente algoritmo di:
Generazione chiave:

1. Selezioniamo casualmente $d \in [1, n-1]$;

2. Calcoliamo $Q = dP$.

Una volta determinata, dobbiamo verificare che la chiave sia valida. Lo scopo della validazione della chiave pubblica consiste nel verificare che vengano rispettate determinate proprietà. Una validazione con esito positivo dimostra che la chiave privata esiste, ma questo non garantisce che tale chiave sia stata effettivamente calcolata, nè tanto meno che il proprietario la possieda. La validazione della chiave è importante per il protocollo di *Scambio chiavi Diffie-Hellman*, dove Alice calcola la chiave privata k combinandola con la chiave pubblica di Bob.

Per la **Validazione della chiave**, si verifica che:

1. $Q \neq \infty$
2. $x_q, y_q \in \mathbb{F}_q$
3. Q soddisfi l'equazione della curva E definita dai parametri A e B
4. $nQ = \infty$
5. - Se almeno uno dei controlli precedenti è fallito, l'algoritmo restituisce "curva rifiutata"
   - altrimenti restituisce "curva accettata"

Due dei crittosistemi analizzati nel terzo capitolo sono quelli riguardanti la firma digitale.

Firma digitale ElGamal

Alice vuole firmare un documento. Supponiamo che il documento che vuole firmare sia in formato elettronico, come ad esempio un file del computer. Quello che potremmo fare è digitalizzare la firma di Alice ed applicarla al file contenente il

documento. Però, in questo modo, l'attaccante Eve potrebbe copiare la firma di Alice ed usarla per firmare un suo documento. Quindi, occorre compiere dei passaggi in modo che la firma non possa essere riutilizzata su un altro documento ed inoltre, deve essere possibile verificare l'autenticità della firma e del mittente.
Vediamo come è possibile firmare un documento, utilizzando le curve ellittiche. Per prima cosa, Alice deve stabilire la sua chiave pubblica.

Generazione chiave pubblica: Alice sceglie:

1. una curva ellittica E su un campo finito $\mathbb{F}_q$ in modo tale che il problema del logaritmo discreto sia intrattabile su $E(\mathbb{F}_q)$.

2. un punto $P \in E(\mathbb{F}_q)$ tale che l'ordine N del punto sia un primo abbastanza grande.

3. un intero segreto a e calcola $A = aP$.

4. una funzione $f : E(\mathbb{F}_q) \longrightarrow \mathbb{Z}$.

La chiave pubblica di Alice è $(E,\ \mathbb{F}_q,\ P,\ A,\ f)$ e la chiave privata è a.

Osserviamo, prima di procedere con l'algoritmo di firma, che la funzione f scelta da Alice, non richiede particolari proprietà, ad eccezione del fatto che la sua immagine dovrebbe essere grande e solo un numero piccolo di input dovrebbe produrre un eventuale output. Ad esempio, se $\mathbb{F}_q = \mathbb{F}_p$, Alice può scegliere la funzione $f(x, y) = x$, dove x è visto come un intero appartentente all'insieme $\{0, \ldots, p-1\}$. Tale funzione prende in input al più due punti (x, y) e restituisce in output un dato x.

Descriviamo ora l'algoritmo per firmare il documento. Alice:

1. rappresenta il documento come un intero m.

2. sceglie un intero casuale k, tale che $MCD(k, N) = 1$, e calcola $R = kP$.

3. calcola $s \equiv k^{-1}(m - af(R)) \pmod{N}$.

Il messaggio firmato è $(m,\ R,\ s)$. Notiamo che Alice non sta cercando di mantenere segreto il suo messaggio m. Per farlo, dovrebbe utilizzare un algoritmo di cifratura. Adesso Bob, per verificare che la firma sia autentica, deve:

1. scaricare le informazioni pubbliche di Alice.

2. calcolare $V_1 = f(R)A + sR$ e $V_2 = mP$.

3. Se $V_1 = V_2$, dichiara che la firma è valida.

Infatti, se la firma è valida, abbiamo:

$$V_1 = f(R)A + sR = f(R)aP + skP = f(R)aP + (m - af(R))P = mP = V_2$$

avendo utilizzato il fatto che $sk \equiv m - af(R) \pmod{N}$.

Anche in questo caso, se Eve riesce a risolvere il logaritmo discreto, può usare P ed A per trovare il segreto a ed applicare la firma di Alice ad un altro messaggio. Analogamente, Eve può usare P ed R per determinare k.
Pertanto, Alice deve mantenere segreti a e k ed inoltre, deve utilizzare un k differente per ogni messaggio che vuole firmare.

La convinzione generale è che la sicurezza del sistema ElGamal è molto vicina alla sicurezza del logaritmo discreto su $E(\mathbb{F}_q)$. Un metodo più efficiente è quello di utilizzare una funzione Hash H per firmare il messaggio $H(m)$. Una funzione crittografica Hash è una funzione che prende in input un messaggio di lunghezza arbitraria (talvolta il messaggio è costituito da miliardi di bit) e restituisce in output valori di lunghezza fissa (ad esempio, 160 bit).
La funzione Hash deve avere le seguenti proprietà:

1. dato un messaggio m, il valore $H(m)$ può essere calcolcato velocemente.

2. dato y, è computazionalmente impossibile trovare m utilizzando $y = H(m)$.

3. è computazionalmente impossibile trovare due messaggi distinti, m_1 ed m_2, utilizzando $H(m_1) = H(m_2)$.

Con lo stesso algoritmo di prima, Alice utilizza la funzione Hash per firmare il messaggio m, dove $(H(m),\ R_H,\ s_H)$ sarà la firma valida, e Bob verifica l'autenticità della firma.

Il vantaggio in questo caso, è che un messaggio molto lungo, contenente miliardi di bit, ha una firma che richiede solo poche migliaia di bit aggiuntivi. Dunque, fintanto che il problema del logaritmo discreto è intrattabile su $E(\mathbb{F}_q)$, Eve non potrà mettere la firma di Alice su un altro messaggio.

ECDSA

Nel 1992 fu proposto, per la prima volta da Scott Vanstone, l'**Elliptic Curve Digital Signature Algorithm** (ECDSA), il quale offre una variante del Digital Signature Algorithm (DSA) utilizzando la crittografia ellittica.
Alice vuole firmare un documento m e per farlo sceglie dei parametri di dominio che utilizzerà per la generazione della sua chiave pubblica. Pertanto, Alice sceglie:

1. una curva ellittica E su un campo finito $\mathbb{F}_q$ tale che $\#E(\mathbb{F}_q) = fr$, dove r è un primo abbastanza grande ed f un intero piccolo (generalmente, si sceglie f uguale a 1, 2 o 4 in modo da rendere efficiente l'algoritmo).

2. un punto base $P \in E(\mathbb{F}_q)$ di ordine r.

3. un intero segreto a e calcola $A = aP$.

Dunque, le informazioni pubbliche di Alice sono $(\mathbb{F}_q,\ E,\ r,\ P,\ A)$ e la chiave privata è a.

Per firmare il messaggio m, Alice:

1. sceglie un intero casuale k, con $1 \leq k < r$ e calcola $R = kP = (x_R, y_R)$.
2. calcola $s \equiv k^{-1}(m + ax_R) \pmod{r}$.

Il documento firmato è $(m,\ R,\ s)$.

Per verificare la validità della firma, Bob:

1. calcola $b = s^{-1}m$ e $c = s^{-1}x_R$ modulo r.
2. calcola $V = bP + cA$.
3. Se $V = R$, allora la firma è valida.

Infatti, se il messaggio è firmato correttamente, si ottiene:

$$V = bP + cA = s^{-1}mP + s^{-1}x_RA = s^{-1}(mP + x_RaP) = kP = R.$$

La principale differenza tra l'ECDSA e lo schema di firma digitale ElGamal, descritto nel paragrafo precedente, sta nella procedura di verifica. Nel sistema ElGamal, l'equazione di verifica richiedeva di effettuare tre volte il calcolo di un intero per un punto, mentre nell'ECDSA tale calcolo si effettua solo due volte. Pertanto, nel caso in cui sono richieste più verifiche di validazione della firma, è preferibile utilizzare l'ECDSA piuttosto che il sistema ElGamal.

Confronto con DSA

Ad un grado di sicurezza di 80 bit (ciò vuole dire che un possibile attacco, richiete la generazione di 2^{80} firme per trovare la chiave privata), la dimensione di una chiave pubblica DSA è all'incirca 1024 bit, mentre la dimensione della chiave ECDSA è di

160 bit. Per quanto riguarda la firma, la dimensione è la stessa: $4t$ bit, dove t è il livello di sicurezza misurato in bit.

Una volta approfonditi sia gli aspetti teorici che implementativi dell'uso delle curve ellittiche in crittografia, effettueremo un confronto tra la sicurezza garantita da questo tipo di crittografia con quella relativa ai problemi della fattorizzazione degli interi e del logaritmo discreto. Poichè le differenze tra i sistemi utilizzati sono tante, per il confronto ci siamo basati sulla dimensione della chiave a parità di sicurezza. Per prima cosa, abbiamo osservato che, a seconda delle esigenze di sicurezza, si possono impiegare algoritmi crittografici differenti, ognuno dei quali può essere implementato con chiavi di dimensioni diverse. In base alle circostanze richieste, una chiave eccessivamente grande può influire negativamente sulle prestazioni dell'algoritmo, mentre l'utilizzo di chiavi molto piccole non ci garantisce un livello elevato di sicurezza. Dunque è opportuno individuare il valore minimo per la dimensione della chiave in modo da garantire un grado di sicurezza sufficiente.

Un altro fattore che abbiamo utilizzato per il confronto è il cosidetto *running-time* (tempo di esecuzione). A livello di complessità, i problemi della fattorizzazione di interi e del logaritmo discreto sono principalmente equivalenti, pertanto è possibile confrontarne anche il grado di sicurezza. Invece, il problema del logaritmo discreto su curve ellittiche è molto più complesso rispetto ai precedenti. Ciò significa che, a parità di lunghezza della chiave, i crittosistemi basati su curve ellittiche sono molto più sicuri rispetto ai sistemi RSA/DSA. Pertanto, per garantire lo stesso grado di sicurezza, i crittosistemi basati su curve ellittiche possono utilizzare chiavi molto più piccole.
Tutte queste osservazioni fatte per la chiave, ci permettono di confrontare anche le dimensioni della firma digitale, ottenendo le stesse considerazioni fatte per la chiave pubblica.

Attualmente, i crittosistemi a chiave pubblica tradizionali (RSA, DSA, Diffie-Hellman) sono senza dubbio i più utilizzati e la diffusione della crittografia basata su curve ellittiche è ancora marginale, ma si può prevedere che l'utilizzo delle curve ellittiche in crittografia si diffonderà in maniera crescente grazie al fatto che:

- la diffusione di dispositivi, che richiedono un elevato grado di sicurezza, è in continuo aumento;
- i requisiti di sicurezza sono sempre più rigorosi;
- all'aumentare del livello di sicurezza richiesto, cresce l'efficienza dei sistemi basati su curve ellittiche rispetto ai sistemi tradizionali.

In attesa di nuovi sviluppi crittografici, le prospettive per l'utilizzo delle curve ellittiche in crittografia sono molto promettenti.

Bibliografia

[1] Henri Cohen and Gerhard Frey, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Chapman and Hall/CRC, 2006.

[2] Catello A. De Rosa, *Sistemi di cifratura. Storia, principi, algoritmi e tecniche di crittografia*, Maggioli Editore, 2010.

[3] Darrel Hankerson, Alfred Menezes, Scott Vanstone, *Guide to Elliptic Curve Cryptography*, Springer, 2004.

[4] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, *An introduction to Mathematical Cryptography*, Springer, 2008.

[5] Richard A. Mollin, *RSA and Public-Key Cryptography*, Chapman and Hall/CRC, 2003.

[6] Lawrence C. Washington, *Elliptic Curves: Number Theory and Cryptography*, Chapman and Hall/CRC, 2nd edition, 2008.

[7] Wikipedia, *Firma digitale*, `https://it.wikipedia.org/wiki/Firma_digitale`, (ultima modifica il 23 settembre 2017).

[8] Wikipedia, *Funzione crittografica Hash*, `https://it.wikipedia.org/wiki/Funzione_crittografica_di_hash` (ultima modifica il 10 settembre 2017).