



Post-Quantum Cryptography Standards

Spiegazione e Implementazione del KEM CRYSTALS-Kyber

Sicurezza dei Sistemi e delle Reti Informatiche

Samuele Manclossi (09882A)

April 24, 2024



UNIVERSITÀ
DEGLI STUDI
DI MILANO



*I recenti miglioramenti nella tecnologia del **Quantum Computing** hanno accelerato la ricerca di soluzioni alternative alla fattorizzazione come **problema difficile**, ricerca già iniziata con la pubblicazione dell'**Algoritmo di Shor**.*

*Questa ricerca si è tramutata tra Agosto e Dicembre 2016 in una gara del **NIST** per ottenere dei nuovi standard. Nel 2017 è stato proposto il sistema che vedremo e nel **2021** esso è diventato formalmente un **nuovo standard**.*



Table of Contents

1 Key Exchange and Quantum Security

► Key Exchange and Quantum Security

► CRYSTALS-Kyber

► Baby-Kyber KEM

► Personalization

► Summary



Un po' di storia

Come siamo arrivati qui

- 1994: Peter Shor pubblica un algoritmo in grado di rompere RSA e Diffie-Hellman fattorizzando in primi in un modo impensabile per i computer classici
- 2005: Oded Regev pubblica una ricerca sul problema del Learning With Errors
- 2016: Il NIST richiede nuovi standard in grado di resistere all'avvento del quantum computing
- 2017: Viene proposto CRYSTALS-Kyber (KEM)
- 2018: Oded Regev vince il Premio Gödel per la sua ricerca
- 2021: CRYSTALS-Kyber è l'unico KEM reso standard
- Esso è ora usato in varie applicazioni tra cui Signal, WhatsApp, un branch di OpenSSL e altri.



Algoritmo di Shor

Fattorizzazione quantistica

- La sicurezza di molti sistemi di crittografia asimmetrica o di scambio di chiavi si basa sulla difficoltà della fattorizzazione
- La difficoltà con l'algoritmo classico migliore che ci sia noto è al momento di

$$O\left(e^{1.9(\log N)^{1/3}(\log \log N)^{2/3}}\right)$$

- L'algoritmo di Shor, pubblicato nel 1994, è in grado di fattorizzare un numero in solo

$$O((\log N)^2(\log \log N))$$

- Per ora non ancora applicabile su grandi numeri
- I computer quantistici continuano a migliorare



Lattice-based Cryptography

Cosa è un lattice

Lattice

Un lattice è, in geometria e nella teoria dei gruppi, un insieme infinito di punti in uno spazio vettoriale tale che la somma o sottrazione delle coordinate di due punti nel lattice produce le coordinate un terzo punto, sempre appartenente al lattice.

Alcune costruzioni basate sui lattice sembrano, al momento, resistenti ad attacchi da parte di calcolatori non solo classici ma anche quantistici!



Lattice-based Cryptography

Learning With Errors

Learning with Errors

Il problema del Learning with Errors (LWE) è un problema matematico basato sull'idea di rappresentare le informazioni come un insieme di equazioni aggiungendo del rumore, ossia degli errori.

Esso è stato introdotto da Oded Regev nel suo lavoro del 2005 ed è stato dimostrato avere la stessa complessità di molti problemi worst-case aventi a che fare con i lattici.



Table of Contents

2 CRYSTALS-Kyber

► Key Exchange and Quantum Security

► CRYSTALS-Kyber

► Baby-Kyber KEM

► Personalization

► Summary



Public Key Encryption Scheme

2 CRYSTALS-Kyber

Per avvicinarsi al KEM occorre costruire delle primitive da poter usare in seguito. Queste sono quelle tipiche della crittografia a chiave pubblica ossia

- Key Generation
- Encryption
- Decryption



Key Generation

2 CRYSTALS-Kyber

La generazione delle chiavi avviene nel seguente modo:

Kyber.CPA.KeyGen():

```
1  $\rho, \sigma \leftarrow \{0, 1\}^{256}$   
2  $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\rho)$   
3  $(\mathbf{s}, \mathbf{e}) \sim \beta_\eta^k \times \beta_\eta^k := \text{Sam}(\sigma)$   
4  $\mathbf{t} := \text{Compress}_q(\mathbf{A}\mathbf{s} + \mathbf{e}, d_t)$   
5  $\text{return}(pk := (\mathbf{t}, \rho), sk := \mathbf{s})$ 
```

- prendo due valori casuali ρ, σ da 256 bit
- genero una matrice \mathbf{A} di dimensioni $k \times k$ con coefficienti modulo q a partire da ρ
- ottengo i valori di chiave privata \mathbf{s} ed errore \mathbf{e} come array di dimensione k con coefficienti piccoli (modulo η) a partire da σ
- ottengo \mathbf{t} come compressione di $\mathbf{A}\mathbf{s} + \mathbf{e}$
- restituisco la coppia chiave pubblica, chiave privata



Encryption

Parte A

Kyber.CPA.Enc(pk m):

```
1  $r \leftarrow \{0, 1\}^{256}$   
2  $\mathbf{t} := \text{Decompress}_q(\mathbf{t}, d_t)$   
3  $\mathbf{A} \sim R_q^{k \times k} := \text{Sam}(\rho)$   
4  $(\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2) \sim \beta_\eta^k \times \beta_\eta^k \times \beta_\eta := \text{Sam}(\mathbf{r})$   
5  $\dots$ 
```

- creo un valore casuale di inizializzazione r
- ottengo \mathbf{t} dal parametro compresso ottenuto in input ($pk = (\mathbf{t}, \rho)$)
- mi ricostruisco la matrice \mathbf{A} usando una Extendable Output Function a partire sempre da ρ (si ricorda che le funzioni Sam sono deterministiche)
- ottengo $\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2$ come due array di polinomi e un polinomio con coefficienti *piccoli*



Encryption

Parte B

Kyber.CPA.Enc(pk m):

```
4  ...  
5   $u := \text{Compress}_q(\mathbf{A}^T \mathbf{r} + \mathbf{e}_1, d_u)$   
6   $v := \text{Compress}_q(\mathbf{t}^T \mathbf{r} + e_2 + \lceil \frac{q}{2} \rceil \cdot m, d_v)$   
7  return  $c := (\mathbf{u}, v)$ 
```

- calcolo \mathbf{u} usando gli errori vettoriali generati precedentemente
- calcolo v comprimendo il risultato della somma di rumore da due fonti $(\mathbf{t}^T \mathbf{r}, e_2)$ e il messaggio appositamente amplificato in modo che abbia coefficienti grandi
- restituisco il ciphertext, ossia la coppia (\mathbf{u}, v)



Decryption

2 CRYSTALS-Kyber

Kyber.CPA.Dec(sk c):

```
1  $\mathbf{u} := \text{Decompress}_q(\mathbf{u}, d_u)$   
2  $\mathbf{v} := \text{Decompress}_q(\mathbf{v}, d_v)$   
3 return  $\text{Compress}_q(\mathbf{v} - \mathbf{s}^T \mathbf{u}, 1)$ 
```

- ottengo \mathbf{u} e \mathbf{v} decomprimendo il ciphertext
- ricavo il plaintext come $\mathbf{v} - \mathbf{s}^T \mathbf{u}$ e approssimando: se il valore del coefficiente è più vicino a 0 o q che a $\lceil \frac{q}{2} \rceil$ allora avrò uno 0, altrimenti un 1



Messaggi e polinomi

2 CRYSTALS-Kyber

Noi siamo abituati a trasmettere i messaggi nella forma di numeri, tuttavia qui lavoriamo con i polinomi. Come è possibile questo?

I numeri sono polinomi

Ci basta effettuare una semplice conversione prendendo il messaggio in binario e associando ad ogni bit un esponente, ad esempio $11_{10} = 1101_2 = x^3 + x + 1$.

Amplificazione del polinomio

Per evitare che i rumori introdotti come parte fondamentale della crittografia vadano a sovrascrivere il nostro messaggio rendendolo inintelligibile amplifichiamo il nostro messaggio di una quantità pari alla metà di q , così lo riusciamo a distinguere. Questo risulta evidente dallo pseudocodice dell'encryption.



Correttezza di Kyber.CPA

2 CRYSTALS-Kyber

Correttezza

Kyber.CPA è $(1 - \delta)$ –corretto con un $\delta < 2^{-128}$. Questo significa che una piccola quantità di decryption potrebbero non portare al valore realmente desiderato. Questa quantità è però trascurabile.



Costruire un KEM

Premessa

Per costruire un KEM dovremo usare le primitive di Key Exchange costruite prima per ottenere due funzionalità:

- Encaps: questa funzione ci fornirà la chiave da proporre e un ciphertext avendo a disposizione una chiave pubblica
- Decaps: questa funzione prende in input una chiave privata e un ciphertext e ottiene la chiave concordata

Esso ovviamente necessita anche di KeyGen, che tuttavia rimane identico a prima



Funzioni di Hashing

2 CRYSTALS-Kyber

Per eseguire alcune operazioni nell'incapsulamento e decapsulamento si fa uso di funzioni di hashing. Seguendo l'articolo **CITALO**, noi useremo due varianti Keccak, ossia sha3-256 come H e sha3-512 come G , il cui output sarà trasformato in polinomi seguendo le solite regole.



Table of Contents

3 Baby-Kyber KEM

- ▶ Key Exchange and Quantum Security
- ▶ CRYSTALS-Kyber
- ▶ **Baby-Kyber KEM**
- ▶ Personalization
- ▶ Summary



Una versione ridotta

Un esempio concreto e una implementazione

Avviso

Per vedere un esempio ridotto fare riferimento a **TODO quote baby Kyber**. L'uso di parametri così bassi potrebbe aumentare il tasso di errori. Inoltre, questo sistema non fa uso di compressione, a differenza del vero CRYSTALS-Kyber.

Qui non sarà riportata questa versione quanto una spiegazione dell'implementazione specifica realizzata come componente chiave del sistema di chat LightKnife.



Table of Contents





4 Personalization

- ▶ Key Exchange and Quantum Security
- ▶ CRYSTALS-Kyber
- ▶ Baby-Kyber KEM
- ▶ **Personalization**
- ▶ Summary



Changing Slide Style

4 Personalization

- You can select the white or *maincolor* **slide style** in the preamble with `\themecolor{white}` (default) or `\themecolor{main}`
 - You should *not* change these within the document: Beamer does not like it
 - If you *really* must, you may have to add `\usebeamercolor[fg]{normal text}` in the slide
- You can change the **footline colour** with `\footlinecolor{color}`
 - Place the command *before* a new frame
 - There are four “official” colors:  `maincolor`,  `stataleyellow`,  `statalegreen`,  `stataledarkgreen`
 - Default is no footnote; you can restore it with `\footlinecolor{}`
 - Others may work, but no guarantees!
 - Should *not* be used with the `maincolor` theme!



Blocks

4 Personalization

Standard Blocks

These have a color coordinated with the footline (and grey in the blue theme)

```
\begin{block}{title}  
content...  
\end{block}
```

Colour Blocks

Similar to the ones on the left, but you pick the colour. Text will be white by default, but you may set it with an optional argument.

```
\begin{colorblock}[black]{statalelightgreen}{title}  
content...  
\end{colorblock}
```

The “official” colours of colour blocks are:



statalelilla,  maincolor,











stataledarkgreen, and  stataleyellow.



Using Colours

4 Personalization

- You can use colours with the `\textcolor{<color name>}{text}` command
- The colours are defined in the `statale_colors` package:
 - Primary colours:  `maincolor` and its sidekick  `statalegrey`
 - Three shades of green:  `statalelightgreen`,  `statalegreen`,  `stataledarkgreen`
 - Additional colours:  `stataleyellow`,  `statalered`,  `statalelilla`
- Do *not* abuse colours: `\emph{}` is usually enough
- Use `\alert{}` to bring the focus somewhere



Using Colours

4 Personalization

- You can use colours with the `\textcolor{<color name>}{text}` command
- The colours are defined in the `statale_colours` package:
 - Primary colours:  `maincolor` and its sidekick  `statalegrey`
 - Three shades of green:  `statalelightgreen`,  `statalegreen`,  `stataledarkgreen`
 - Additional colours:  `stataleyellow`,  `statalered`,  `statalelilla`
- Do *not* abuse colours: `\emph{}` is usually enough
- Use `\alert{}` to bring the focus somewhere
- If you highlight too much, you don't highlight at all!



Adding images

4 Personalization

Adding images works like in normal \LaTeX :

Code for Adding Images

```
\usepackage{graphicx}  
% ...  
\includegraphics[width=\textwidth]  
{assets/logo_RGB}
```





Splitting in Columns

4 Personalization

Splitting the page is easy and common; typically, one side has a picture and the other text:

This is the first column

And this the second

Column Code

```
\begin{columns}
  % adding [onlytextwidth] the left margins will be set correctly
  \begin{column}{0.6\textwidth}
    This is the first column
  \end{column}
  \begin{column}{0.3\textwidth}
    And this the second
  \end{column}
  % There could be more!
\end{columns}
```



Special Slides

4 Personalization

- Chapter slides
- Side-picture slides










UNIVERSITÀ
DEGLI STUDI
DI MILANO



Chapter slides

4 Personalization

- Similar to `frames`, but with a few more options
- Opened with `\begin{chapter} [<image>] {<color>} {<title>}`
- Image is optional, colour and title are mandatory
- There are seven “official” colours:  `maincolor`,  `stataledarkgreen`,  `statalegreen`,  `statalelightgreen`,  `statalered`,  `stataleyellow`,  `statalelilla`.
 - Strangely enough, these are *more* than the official colours for the footline.
 - It may still be a nice touch to change the footline of following slides to the same color of a chapter slide. Your choice.
- Otherwise, `chapter` behaves just like `frame`.



Side-Picture Slides

4 Personalization

- Opened with
`\begin{sidepic}{<image>}{<title>}`
- Otherwise, `sidepic` works just like `frame`





Fonts

4 Personalization

- The paramount task of fonts is being readable
- There are good ones...
 - Use serif fonts only with high-definition projectors
 - Use sans-serif fonts otherwise (or if you simply prefer them)
- ... and not so good ones:
 - Never use monospace for normal text
 - Gothic, calligraphic or weird fonts should always be avoided



Look

4 Personalization

- To insert a final slide with the title and final thanks, use `\backmatter`.
 - The title also appears in footlines along with the author name, you can change this text with `\footlinepayoff`
 - You can remove the title from the final slide with `\backmatter[notitle]`
- The aspect ratio defaults to 16:9, and you should not change it to 4:3 for old projectors as it is inherently impossible to perfectly convert a 16:9 presentation to 4:3 one; spacings *will* break
 - The `aspectratio` argument to the `beamer` class is overridden by the SINTEF theme
 - If you *really* know what you are doing, check the package code and look for the `geometry` class.



Table of Contents

5 Summary

- ▶ Key Exchange and Quantum Security
- ▶ CRYSTALS-Kyber
- ▶ Baby-Kyber KEM
- ▶ Personalization
- ▶ Summary



Good Luck!

5 Summary

- Enough for an introduction! You should know enough by now
- If you have corrections or suggestions, [send them to me!](#)



Post-Quantum Cryptography Standards

Thank you for listening!
Any questions?