

Biometric Authentication Tag System su piattaforma Android

Alessandro Loffredo - Daniele D'Angeli

January 4, 2012

Contents

1	Introduzione	3
2	Stato dell'arte	3
2.1	Firme Digitali	3
2.2	Face Recognition	3
2.3	QR Code	3
3	Sistema Realizzato	4
3.1	Descrizione	4
3.1.1	Use Case Diagram	4
3.2	Strumenti utilizzati	4
3.2.1	Andorid SDK	4
3.2.2	Andorid NDK	4
3.2.3	Libreria OpenCV	4
3.3	Firme Digitali	4
3.4	Face Recognition	4
3.5	ID Card	4
3.5.1	NFC	4
3.5.2	QRCode I/O	5
4	Testing, Tuning e analisi delle prestazioni	6
5	Conclusioni e sviluppi futuri	6

1 Introduzione

Il seguente lavoro ha l'obiettivo di creare una cartà di identità biometrica, nella quale verranno riportate le caratteristiche facciali di un individuo per permettere il riconoscimento. Il sistema si compone di due fasi: una prima fase permette di estrarre le features facciali, le inserisce in un qrcode con opportuna firma digitale, e crea la carta di identità biometrica. Una seconda fase, permette, a partire dalla carta di identità biometrica, di verificare se il possessore di tale carta sia effettivamente il proprietario di essa, verificando le features facciali e la firma digitale. Il sistema è stato realizzato su dispositivi Android os.

2 Stato dell'arte

2.1 Firme Digitali

Una firma digitale è uno schema matematico per dimostrare l'autenticità di un messaggio, o in generale, di un documento. Di fatto, permette di verificare che il documento non sia stato modificato dopo la creazione, da terze parti illecite. Si utilizzano algoritmi di cifratura con chiave pubblica. L'idea è di firmare, ovvero codificare con la propria chiave privata, un hash del documento ed appenderlo

2.2 Face Recognition

stato arte face recognition

2.3 QR Code

Si tratta di un codice a barre bidimensionale a matrice, composto da moduli neri disposti all'interno di uno schema di forma quadrata. Viene impiegato per memorizzare informazioni generalmente destinate ad essere lette tramite un dispositivo mobile (es. smartphone, tablet).

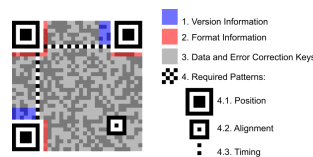


Figure 1: Struttura Qrcode

Un solo crittogramma può contenere fino a 7.089 caratteri numerici o 4.296 alfanumerici. Il loro utilizzo è per lo più

Capacità di memorizzazione
Solo numerico: max 7.089 caratteri.
Alfanumerico: max 4.296 caratteri
Binario (8 bit): max 2.953 byte
Kanji/Kana: max 1.817 caratteri.

Capacità di correzione degli errori
Livello L: circa il 7% delle parole in codice può essere ripristinato.
Livello M: circa il 15% delle parole può essere ripristinato
Livello Q: circa il 25% delle parole può essere ripristinato
Livello H: circa il 30% delle parole può essere ripristinato

3 Sistema Realizzato

3.1 Descrizione

Descrizione sistema

3.1.1 Use Case Diagram

registrazione e verifica

3.2 Strumenti utilizzati

3.2.1 Andorid SDK

3.2.2 Andorid NDK

3.2.3 Libreria OpenCV

3.3 Firme Digitali

3.4 Face Recognition

3.5 ID Card

3.5.1 NFC

descrizione NFC

3.5.2 QRCode I/O

La gestione dei QRCode nel progetto è un aspetto molto importante, che coinvolge e la fase di creazione di una cartà d'identità di un individuo, e la fase di riconoscimento. L'importanza dei QRCode risiede nel fatto che grazie ad essi, è possibile recuperare le features facciali (firmate digitalmente) per confrontarle con le features estratte al momento dell'identificazione.

Di conseguenza risulta essenziale generare un QRCode nella fase di creazione, e poter leggere il suo contenuto nella fase di autenticazione. Per la gestione dei QRCode, è stata utilizzata una libreria OpenSource denominata 'ZXing'¹, la quale permette di eseguire svariate operazioni con i codici a barre. Essendo la libreria Open, sono state estratte soltanto le funzionalità utili al progetto.

Generazione Per la generazione dei QRCode è stata realizzata una classe denominata "QRCodeEncoder" la quale offre dei metodi per generare un QR-Code in formato "Bitmap", a partire da un insieme di dati quali dimensione del QRCode e i "byte" da inserire al suo interno.

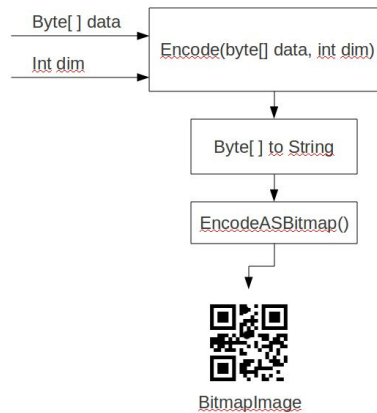


Figure 2: Funzionamento della procedura di creazione del QRCode

Il vettore di byte passato alla funzione, viene convertito in un stringa, per poi essere codificato nel QRCode. In lettura si esegue il processo inverso. La funzione dell'oggetto QRCodeEncoder è utilizzata staticamente nel progetto. Il metodo in questione è denominato *encodeAsBitmap* ed esegue le operazioni appena descritte. Il metodo è utilizzato, richiamandolo staticamente in questo modo: *QRCodeEncoder.encodeASBitmap(byte[]*

¹<http://code.google.com/p/zxing/>

data, in dim). Una volta che viene restituito l'oggetto Bitmap, questo verrà gestito dal programma, in modo tale da allegarlo alla mail che viene inoltrata all'amministratore del sistema.

Lettura La fase di decodifica deve essere effettuata per poter risalire alla features facciali appartenenti ad un individuo. La lettura di un QRCode è effettuata grazie ad una scansione dalla videocamera di un dispositivo mobile. Anche in questo caso, le librerie ZXing offrono delle funzionalità per operare il processo di scansione. A differenza della generazione, la lettura avviene lanciando una Intent esplicita in modalità: *startActivityForResult*². La Intent viene dichiarata e lanciata in questo modo:

```
Intent intent = new Intent("com.google.zxing.client.android.SCAN");
intent.setPackage("com.google.zxing.client.android");
intent.putExtra("SCAN_MODE", "QR_CODE_MODE");
startActivityForResult(intent, 0);
```

In lettura, viene implementato il metodo:

```
public void onActivityResult(int requestCode,
                             int resultCode,
                             Intent intent)
```

, nel quale viene prelevato dalla intent, una stringa extra

```
intent.getStringExtra("SCAN_RESULT")
```

, la quale contiene il risultato della scansione. Questo risultato sarà gestito in modo tale da ottenere il contenuto delle informazioni, ovvero le features facciali, firmate digitalmente.

Il comportamento della lettura e della generazione dei QRCode è il seguente:

4 Testing, Tuning e analisi delle prestazioni

Testing del sistema

5 Conclusioni e sviluppi futuri

Il lavoro effettuato ha portato allo studio

²Su Android è possibile invocare altre activity in varie modalità, tra le quali quella appena descritta. Le Activity lanciate in questo modo offrono un servizio, quindi per poter gestire il risultato deve essere implementato il metodo: *OnActivityResult*

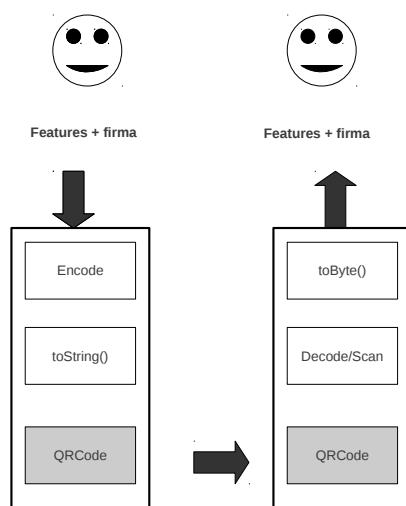


Figure 3: Gestione QRCode fasi generazione - lettura