Inviare pdf in modo sicuro

Roberto Giacomelli
Articolo sul blog *http://robitex.wordpress.com*
e-mail: giaconet dot mailbox at gmail dot com

22 maggio 2011

Sommario

Dopo aver analizzato il problema della sicurezza nella trasmissione di documenti viene descritta nel dettaglio una soluzione semplice e sufficientemente sicura con il programma libero pdftk limitatamente al caso di documenti pdf. In particolare sono riportati i comandi per la cifratura pdf a 128 bit e l'operazione inversa di decifratura per riottenere il pdf in chiaro.

Indice

1 Il problema

Con la diffusione di internet lo scambio di documenti avviene sempre più spesso tramite il servizio base della posta elettronica. Tuttavia i messaggi e-mail non sono sicuri ed i documenti che un tempo venivano consegnati o ritirati a mano con la sicurezza della *fisicità*, oggi possono essere letti da terzi con la violazione dei messaggi digitali che transitano su connessioni internet non protette.

Pensiamo al caso di una piccola azienda od un artigiano che ogni mese riceve nel formato PDF i moduli di pagamento F24 od i cedolini delle buste paga dei propri dipendenti dall'ufficio del commercialista.

> ***Quasi sicuramente il testo del messaggio e-mail conterrà la classica dicitura sulla protezione dei dati personali secondo la legge sulla privacy, ma i documenti PDF allegati saranno invariabilmente in chiaro!!!***

Cosa possiamo fare? Tornare a visitare i vari uffici per raccogliere di persona i documenti cartacei? Saremo costretti a prendere l'auto impiegando tempo ed inquinando ulteriormente l'ambiente urbano, anche se ne gioverebbero i rapporti interpersonali (il computer ci fa vivere in un mondo cerebrale...).

2 Una soluzione

La soluzione più robusta è quella di *blindare* il messaggio di posta elettronica garantendo non solo l'integrità dell'e-mail ma anche la sicurezza sull'identità del mittente. Ci si può rivolgere per esempio a software come GNUPG che fa esattamente quello che serve e di cui potete trovare una splendida guida in italiano di Mario Pascucci sempre valida e completa sia per Windows che per Linux.

Per usare GNUPG è necessario creare una coppia di chiavi, quella pubblica e quella privata, creare una *password phrase* per la sicurezza locale del sistema, utilizzare il sistema di server per l'upload della chiave pubblica, ed accertarsi **fisicamente** che la chiave pubblica del proprio corrispondente appartenga effettivamente a lui. Da quel momento sarà possibile crittografare i messaggi e pure firmarli digitalmente.

Tuttavia questo sistema mi sembra un po' troppo. Direi quindi di sacrificare un po' di sicurezza ma di acquistare semplicità concettuale rinunciando alla certezza del mittente del messaggio.

3 Una soluzione più comoda

Assumendo di voler trasmettere solo file PDF, li blinderemo con l'utility libera pdftk. Ecco quel che dovremo fare: preparare i due PC di mittente e destinatario e costruire una password robusta nota solo ad entrambi e cifrare/decifrare i documenti.

Se solo al mittente ed al destinatario è nota la password e se il destinatario riesce a decifrare correttamente il file pdf contenuto nel messaggio, allora possiamo ragionevolmente supporre che solamente i due soggetti hanno avuto accesso al messaggio (e quindi la comunicazione è sicura), e che il nostro mittente sia effettivamente colui che ha spedito il messaggio. Diversamente vorrà dire che la password sarà stata violata in qualche modo. Dunque è essenziale che la password sia robusta e nota solo agli interessati, altrimenti tutta la procedura verrà inficiata.

3.1 Preparare il PC

Per preparare il PC basta installare pdftk. Assumendo che il sistema sia un pc Ubuntu è sufficiente digitare il comando:

```
1 sudo apt-get install pdftk
```

Anche per gli utenti Windows l'installazione è semplice: è sufficiente seguire la procedura descritta alla pagina ufficiale del pacchetto (si tratta di scaricare un file e scompattarlo in una cartella opportuna).

3.2 Scambiarsi la password

Una volta costruita una password robusta, per esempio 'g456Wsa+dh', annotatela su un foglio e recatevi **di persona** dal vostro corrispondente. Questo passaggio è essenziale perché come già detto, dalla sicurezza di questa password dipenderà la sicurezza di tutti i vostri documenti contenuti nei messaggi e-mail. Se il corrispondente è geograficamente lontano potreste ricorrere al telefono od alla posta, con le modalità fantasiose che si possono inventare divertendosi (per esempio una specie di caccia al tesoro in cui la password viene suddivisa in due o tre tronconi e trasmessa con mezzi fisici diversi assieme alle *istruzioni di montaggio*, ma solo se il corrispondente mostra spirito goliardico).

4 Procedura operativa: cifratura

Sia **mydoc.pdf** il file pdf da trasmettere allora per cifrarlo a 128 bit aprite il Terminale (o la finestra di console di Windows) e digitate il comando:

```
1 pdftk mydoc.pdf output cryptmydoc.pdf owner_pw g456Wsa+dh user_pw IlMioNome
```

Fatto.

Si tratta di una normale sintassi per utility a linea di comando dove gli argomenti separati da spazi seguono il nome del comando da lanciare e sono strutturati in un primo gruppo costituito dal file di input e da un secondo gruppo composto dal nome da dare al file di output con le opzioni, separati dalla con la chiave 'output'. Volendo elencare la sintassi potremo costruire l'elenco degli argomenti:

1 - il nome del file pdf da cifrare (nel nostro esempio mydoc.pdf); 2 - la chiave 'output'; 3 - il nome del file cifrato (tipo 'crypt_mydoc.pdf') al quale va applicata la protezione con password (opzione owner_pw seguita dalla password) e l'attribuzione all'utente (opzione 'user_pw' seguita dal nome utente).

Figura 1: La richiesta della password per aprire il pdf cifrato con pdftk

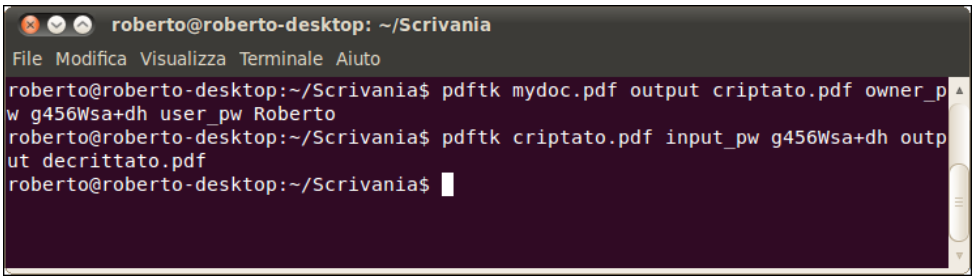

Figura 2: pdftk al lavoro nel terminale di Ubuntu

5 Procedura operativa: decifratura

Dal punto di vista del destinatario, una volta ricevuta l'e-mail con il documento pdf cifrato, potrà o aprirlo direttamente inserendo la password o riportarlo in chiaro per un utilizzo successivo più agevole, con il comando:

```
1 pdftk cryptmydoc.pdf input_pw g456Wsa+dh output mydoc.pdf
```

Finito.

Anche per questo comando un commento: il primo gruppo di argomenti è costituito dal file da decifrare seguito dall'opzione 'input_pw' a cui far seguire la password (altrimenti pdftk non potrà operare sul documento), mentre il secondo gruppo sempre separato dal primo con la chiave 'output', è semplicemente il nome da dare al pdf decifrato.

6 Alcuni screenshot dell'operazione

Ecco cosa viene mostrato se apriamo con Evince, il lettore di pdf predefinito in Ubuntu, un file pdf cifrato:

Nel seguente screenshot potete osservare una sessione di terminale per la cifratura ed una decifratura di un file pdf in Linux Ubuntu:

Nell'immagine seguente invece vi ho riportato il dialogo di proprietà di Acrobat Reader relativo ad un file cifrato con pdftk dove è chiaramente leggibile che ci troviamo davanti un documento blindato con cifratura a 128bit.

Figura 3: Il pdf è proprio blindato!

7 Evoluzione

Naturalmente vi sono modi di rendere il processo più comodo ed user friendly che convengono solo se si hanno un gran numero di file da gestire con destinatari diversi ognuno con password distinte. Penso per esempio ad un utility di spedizione ad interfaccia grafica che sfrutti un database, cifri i pdf, esegua gli invii dei documenti per e-mail e registri le trasmissioni. Con qualcosa del genere l'intera procedura diventerebbe non solo molto più sicura e professIonale ma anche molto più rapida ed efficiente rispetto alle soluzioni manuali caso per caso in chiaro.

Per approfondimenti sulle opzioni di cifratura offerte da pdftk consultate il suo manuale, mentre per il resto potete commentate il post per riportare esperienze e suggerimenti. Bene, direi che siamo giunti al termine per cui vi saluto augurandovi buon lavoro!!!

8 Licenza ed informazioni varie

8.1 Distribuzione/Citazioni

- Inserire nel blockquote solo il primo periodo (le prime poche frasi) di un articolo, diciamo fino ad arrivare al link “Leggi il resto...”;
- Linkare la rimanente parte dell'articolo all'originale su http://robitex.wordpress.com.

Grazie per la collaborazione.

8.2 Colophon

Questo documento è stato composto con LaTeX attraverso uno script in Lua chiamato wp2pdf che elabora il file originale *html* del post pubblicato sul blog in WordPress. Si tratta di una versione migliorata del codice pubblicato sul blog stesso. Per saperne di più contattatemi via posta elettronica all'indirizzo nel titolo del documento, o lasciate un commento sul blog.