

La protezione degli SMS con il modello PGP: firma digitale e cifratura di SMS con la crittografia delle curve ellittiche

Ugo Chirico
<http://www.ugochirico.com>

Abstract—Il bisogno sociale di sicurezza e di privacy nelle comunicazioni digitali è in continua crescita, soprattutto alla luce dei recenti avvenimenti legati alle intercettazioni telefoniche illegali. Cercando di rispondere a tale bisogno, in questo lavoro si presenterà un sistema di protezione delle comunicazioni via SMS, mediante firma digitale e cifratura, che segue il modello PGP e che sfrutta la crittografia delle curve ellittiche come strumento per l'implementazione degli algoritmi crittografici di protezione. In questo lavoro si presenteranno brevemente le principali problematiche di sicurezza legate allo scambio di SMS, evidenziando i motivi per i quali i semplici SMS non sono adatti allo scambio d'informazioni private e non sono adeguati per eseguire transazioni commerciali/bancarie che richiedono l'identificazione certa del mittente e il non ripudio della transazione. Si mostrerà poi come la crittografia delle curve ellittiche sia la tecnologia appropriata per la protezione degli SMS e per l'implementazione del modello PGP sui telefoni cellulari e, infine, si presenterà un'implementazione di tale modello su telefoni cellulari Java J2ME.

Index Terms— SMS, Spoofing, Crittografia, Curve Ellittiche, Cifratura, Firma Digitale, PGP, J2ME, MIDP2.0. WMA1.0

I. INTRODUZIONE

MILIONI di persone si scambiano ogni giorno SMS per comunicare velocemente e a basso costo., ma pochi conoscono le vulnerabilità di questa tecnologia e i rischi di violazione della privacy a cui sono sottoposti. Da un lato, gli operatori telefonici sono tenuti a conservare sui propri server tutti gli SMS in chiaro, analogamente alle trasmissioni vocali, per facilitare le operazioni di polizia mirate ad individuare comunicazioni sospette. Inoltre, gli SMS inviati e ricevuti sono solitamente conservati in chiaro nella SIM o nella memoria del cellulare, quindi basta avere a disposizione per qualche minuto il telefono cellulare per violare la privacy del proprietario. Dall'altro, le specifiche GSM non definiscono un meccanismo per garantire l'autenticità del mittente di un SMS. Difatti, basta avere accesso ad un SMS-gateway per inviare SMS con mittente falso o inesistente (si veda SMS spoofing¹).

¹ L'SMS spoofing consiste nell'invio di SMS il cui mittente è falso o inesistente. Solitamente, gli operatori di telefonia mobile offrono un servizio d'invio SMS tramite appositi SMS-gateway raggiungibili via TCP-IP, mediante il quale si riesce ad inviare SMS con mittente fasullo. Per un

In definitiva, quindi, allo stato attuale gli SMS non sono adatti a trasmettere dati sensibili (come, ad esempio, informazioni segrete o personali) o per trasmettere informazioni che richiedono l'identificazione certa del mittente (disposizioni bancarie, commerciali e simili) e il non-ripudio della trasmissione.

L'obiettivo di questo lavoro è realizzare un sistema di protezione delle comunicazioni via SMS che risolve i problemi sopra menzionati, sfruttando le funzioni di firma digitale e di cifratura secondo il modello PGP, ma usando la crittografia delle curve ellittiche come strumento per l'implementazione degli algoritmi crittografici di protezione.

II. SMS "CRITTOGRAFATI"

Come definito nelle specifiche GSM un SMS testuale può contenere al massimo 160 caratteri ASCII (140 byte se binario²). Qualsiasi strumento di protezione degli SMS deve tenere conto di questo limite e operare di conseguenza.

L'algoritmo a chiave pubblica attualmente più diffuso è l'RSA solitamente usato con chiave da 1024 bit (anche se ormai si sta passando a chiavi da 2048 poiché quelle da 1024 sono ormai poco sicure). Per com'è definito, l'algoritmo RSA, usato secondo le specifiche PKCS#1, produce blocchi crittografati (cifrati o firmati) da 128 byte (ovvero 1024 bit) qualunque sia la lunghezza dell'input. Inoltre, la lunghezza di una chiave pubblica è compresa tra 129 e 256 byte (lunghezza del modulo + lunghezza dell'esponente pubblico).

Il confronto tra questi valori e i limiti di lunghezza imposti sugli SMS ci mostra che l'algoritmo RSA non è appropriato per realizzare un'applicazione PGP-like per SMS. Difatti, un messaggio cifrato con RSA contiene la chiave di sessione (simmetrica) cifrata con la chiave pubblica del destinatario, seguita dal messaggio cifrato vero e proprio. Pertanto la lunghezza effettiva in byte di un SMS trattato con RSA 1024 sarebbe: $128 + Len(ciphertext)$, dove $Len(ciphertext)$ esprime la lunghezza in byte del messaggio cifrato (che corrisponde ad

approfondimento sull'argomento si veda: De Santis A., Spoofing", http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0102/Spoofing_Slide.pdf

² La lunghezza effettiva in byte di un SMS è 140 byte. La particolare codifica a 7 bit definita nelle specifiche GSM però, consente di riportare in 140 byte ben 160 caratteri dell'alfabeto occidentale.

un multiplo della lunghezza del blocco caratteristica dell'algoritmo di cifratura simmetrico usato, ad es. DES, 3DES, AES, ecc.). Senza considerare poi che le operazioni in modulo caratteristiche dell'RSA richiedono notevoli risorse computazionali che spesso non sono disponibili sui telefoni cellulari.

Si è quindi passati a considerare la crittografia delle curve ellittiche come possibile sostituto di RSA.

La crittografia delle curve ellittiche ha diverse caratteristiche molto gradevoli per trattare SMS. In primo luogo, a parità di sicurezza, consente di avere chiavi di lunghezza notevolmente inferiore a quelle RSA. Facendo riferimento alla Fig.1, una chiave ellittica con soli 160 bit offre lo stesso livello di sicurezza di una chiave RSA da 1024 bit.

<i>Time to break in MIPS years</i>	<i>RSA/DSA key size</i>	<i>ECC key size</i>	<i>RSA/ECC key size ratio</i>
10^1	512	106	5 : 1
10^4	768	132	6 : 1
10^{11}	1,024	160	7 : 1
10^{23}	2,048	210	10 : 1
10^{73}	21,000	600	35 : 1

Fig. 1 Confronto tra RSA e Curve Ellittiche (ECC)

In secondo luogo, le operazioni matematiche richieste dagli algoritmi caratteristici delle curve ellittiche sono molto più semplici e pertanto non richiedono grandi risorse di calcolo.

La crittografia delle curve ellittiche quindi, è risultata essere la tecnologia crittografica più adeguata a proteggere le comunicazioni via SMS.

III. IMPLEMENTAZIONE

Per l'implementazione è stata scelta la curva³ definita su *Prime Finite Field* GF(p) con $\|p\| = 192$:

A. Cifratura

Nella versione attuale, la cifratura degli SMS avviene in due fasi: 1) generazione della chiave di sessione mediante l'algoritmo di derivazione delle chiavi ECSVDP-DH descritta in [2]; 2) cifratura del messaggio mediante l'algoritmo AES256 [3] usando la chiave generata al punto 1). Poiché la chiave simmetrica generata con ECSVDP-DH, applicata alla chiave privata del mittente e alla chiave pubblica del destinatario, non deve essere condivisa con il destinatario (che può calcolarla usando lo stesso algoritmo), la lunghezza complessiva dell'SMS cifrato corrisponde alla lunghezza del solo testo cifrato che, in byte, risulta essere un multiplo intero di 32 (256 bit di AES256) quindi, un messaggio in chiaro da 120 caratteri diventa 128 byte in cifra.

³ Le curve ellittiche da usare a scopi crittografici sono definite e standardizzate nel documento di specifiche standard X9.62 [1]. Per questo lavoro è stato utilizzata la curva *prime192v3*.

B. Firma digitale

La firma digitale viene eseguita secondo l'algoritmo ECSP-DSA descritto in [2].

Considerando che un SMS firmato è composto dal testo in chiaro seguito dalla firma digitale, la lunghezza complessiva diventa pari a: $Len(\text{testo}) + 2 * \|p\|$ (espresso in byte)

Quindi, utilizzando $\|p\| = 192$ si ha che la lunghezza in byte di un SMS firmato è pari a $48 + Len(\text{testo})$.

C. Verifica della firma digitale

La verifica di un SMS firmato avviene mediante l'algoritmo ECVP-DSA descritto in [2].

D. Scambio delle chiavi pubbliche

Lo scambio delle chiavi pubbliche può avvenire mediante l'invio di un SMS. Una chiave pubblica ellittica è un particolare punto sulla curva e può essere rappresentato mediante le due coordinate X e Y, ciascuna di lunghezza in bit pari a $\|p\|$. Sfruttando una particolare proprietà delle curve ellittiche, un punto può essere rappresentato anche in forma compressa mediante la sola coordinata X e il bit più significativo della coordinata Y. Quindi, con $\|p\| = 192$ una chiave pubblica può essere rappresentata in forma compressa con 193 bit (ovvero con soli 25 byte).

IV. REALIZZAZIONE SU TELEFONI J2ME

Una volta definiti tutti gli aspetti relativi all'implementazione degli algoritmi crittografici e allo scambio delle chiavi pubbliche, si è passati alla realizzazione di un prototipo per telefoni cellulari Java J2ME.

Il requisito fondamentale è il supporto delle specifiche MIDP2.0 e WMA1.0 [4] per l'invio/ricezione di SMS da parte della piattaforma Java disponibile sul cellulare.

Il prototipo, battezzato "*Message in a Bottle*", offre le funzionalità tipiche di un'applicazione PGP-like, quali invio di un SMS cifrato e/o firmato, invio della propria chiave pubblica con un SMS, un Key-Ring nel quale sono memorizzate le chiavi pubbliche degli interlocutori, ciascuna associate al numero di telefono e, infine, una rubrica degli SMS inviati/ricevuti protetta mediante cifratura con AES256 usando come *masterkey* un PIN scelto dall'utente in fase d'installazione.

Il prototipo è disponibile per il download, in versione dimostrativa, al seguente indirizzo: <http://www.ugosweb.com>

V. CONCLUSIONI

Il lavoro presentato in quest'articolo può essere la base per la realizzazione di servizi a valore aggiunto che sfruttano il canale SMS quale mezzo di trasmissione sicura (servizi di m-commerce e/o m-banking). Il prototipo realizzato è solo un esempio d'applicazione che mostra le funzionalità del motore crittografico a curve ellittiche. Tale motore è un modulo a se stante che fornisce una sorta di API atta a realizzare applicazioni di vario tipo che richiedono trasmissioni sicure con SMS.

BIOBLOGRAFIA

- [1] AMERICAN NATIONAL STANDARD X9.62-1998, Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)©
- [2] IEEE P1363, Standard Specifications for Public Key Cryptography
- [3] Advanced Encryption Standard, Wikipedia
http://it.wikipedia.org/wiki/Advanced_Encryption_Standard

Ugo Chirico si è laureato in Fisica Cibernetica all'università di Napoli Federico II e da diversi anni si occupa di sicurezza applicativa ed in particolare di Crittografia e Smart Card. E' autore del testo "Programmazione delle Smart Card", edito da Infomedia e di numerosi articoli pubblicati sulle riviste "Computer Programming" e "Dev" e presentati a vari convegni internazionali.

E' inoltre esperto di progettazione e sviluppo software in C++, Java, .NET e Prolog e recentemente ha cominciato ad occuparsi di telefonia mobile ed in particolare di applicazioni per telefoni cellulari su piattaforme J2ME, Symbian e Windows Mobile.