SETEFI

INTRODUZIONE AL PROGETTO

Il nostro obiettivo è quello di illustrare la struttura e le caratteristiche di fondo che stanno alla base delle transazioni online operate tramite Setefi, società del gruppo Intesa Sanpaolo che gestisce i pagamenti con moneta elettronica. Cominceremo con lo spiegare il protocollo SSL, utilizzato per scambiare messaggi in sicurezza, per poi passare a mostrare come effettivamente opera Setefi. Nell'ultima parte del seminario parleremo di un esempio di applicazione che permette le operazioni di cifratura e decifratura delle varie informazioni riservate che Setefi scambia con i propri negozianti online.

PARTE 1: SSL

SSL e' un protocollo aperto e non proprietario introdotto da Netscape Communication nel Dicembre 1994, e si e' imposto come standard de facto negli anni successivi. La versione 3.0 sviluppata nel 1996, evoluzione della 2.0 del 1994, e' stata sottoposta all'IETF (Internet Engineering Task Force) per la standardizzazione. Il futuro di SSL e' rappresentato dal protocollo TLSv1 (considerato come SSL 3.1) standardizzato nel 1998.

1.CARATTERISTICHE

1.1 Funzionalità

La comunicazione in rete tra due host solitamente non avviene in maniera diretta, ma coinvolge tutta una serie di sistemi di computer che funzionano da intermediari nel trasporto dei dati. SSL si occupa di assicurare che questi sistemi intermedi non interferiscano nella sessione o acquisiscano informazioni confidenziali; questo protocollo, cioè, garantisce confidenzialità, integrità e autenticazione (mentre, come vedremo, può creare un problema di disponibilità).

Più in dettaglio il protocollo fornisce le seguenti caratteristiche:

Autenticazione: E' il processo grazie al quale si e' in grado di stabilire l'identità dell'interlocutore, garantendo alle parti in causa di essere in comunicazione con entità fidate. I due host si scambiano certificati di identità, la cui validità e' sottoscritta dalle Certification Authorities (CA). In un modello di comunicazione client-server, SSL permette l'autenticazione dell'uno e dell'altro. Il protocollo prevede anche la possibilità di comunicare in forma anonima, chiaramente in questo caso è meno sicuro.

***Privatezza del collegamento*:** i dati sensibili scambiati sul canale di comunicazione sono protetti utilizzando algoritmi di crittografia a chiave

simmetrica, in cui cioè la stessa chiave è utilizzata sia in fase di cifratura dei dati che in fase di decodifica. SSL mette a disposizione più algoritmi di cifratura simmetrica (DES, RC4, ecc.), con diversi livelli di sicurezza, permettendo una qualità del servizio adeguata alla sensibilità dei dati.

Integrità delle informazioni: i dati, prima di essere inviati, vengono autenticati mediante un campo Message Autentication Code (MAC), generato mediante le funzioni hash di firma (MD5, SHA, ecc.) che SSL mette a disposizione.

1.2 Stati

Dovendo stabilire una sessione SSL, le parti che entrano in gioco nella comunicazione devono essere in grado di accordarsi inizialmente su tutta una serie di informazioni comuni, per poter essere in grado di gestire un qualche scambio di messaggi cifrato. Le informazioni concordate devono poi essere memorizzate dalle due parti: l'insieme di tutte le metodologie e dati riservati su cui gli host si sono accordati in fase preliminare prende il nome di stato. Uno stato è quindi una struttura dati che raccoglie tutti gli elementi che identificano una comunicazione.
In una comunicazione tramite SSL si identificano due stati paralleli e consistenti per ambo le parti:

***stato di sessione*:** tale stato viene creato nella fase di handshake e rimane valido per tutta la durata di una sessione SSL. Contiene i seguenti parametri:

Session ID: e' un identificativo scelto dal server per una particolare sessione SSL;

compression_method: specifica l'algoritmo di compressione da utilizzare;

cipher_spec: specifica l'algoritmo di crittografia simmetrico (DES,RC2 etc...) e l'algoritmo per il calcolo del MAC.

peer certificate: certificato dell' host remoto X.509

master_secret: e' una sequenza segreta di 48 bytes dalla quale vengono generate tutte le chiavi di cifratura.

is_resumable: indica se tale sessione e' riesumabile. Se la flag is_resumable e' attiva, la sessione SSL in atto può includere più connessioni in tempi diversi.

***stato di connessione*:** si inizializzano e memorizzano tutte le chiavi utilizzate nei processi crittografici e i dati pseudocasuali in grado di fornire il giusto quantitativo di entropia agli algoritmi di hashing utilizzati per l'autenticazione. Possono esistere più connessioni per ogni sessione. Contiene i seguenti parametri:

server_random: e' una sequenza di bytes casuali scelta dal server.

client_random: e' una sequenza random di numeri scelta dal client di dimensione pari a server_random.

server_write_MAC_secret: e' la chiave segreta nota solo al server utilizzata nelle operazioni di generazione del MAC.

client_write_MAC_secret: e' la chiave segreta nota sola al client con le

stesse funzionalita' di quella per il server.

server_write_key e *client_read_key:* sono chiavi identiche note ad ambo le parti. Vengono utilizzate nei processi di crittografia simmetrica da parte del server verso il client.

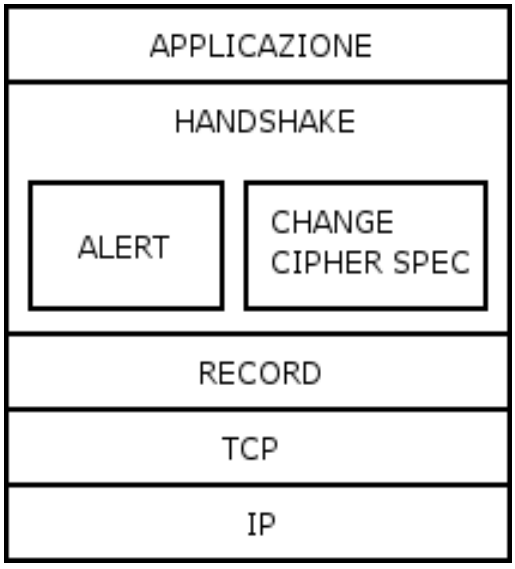
client_write_key e *server_read_key:* sono chiavi identiche note ad ambo le parti. Vengono utilizzate nei processi di crittografia simmetrica da parte del client verso il server.

sequence_number: rappresenta il numero di sequenza per una particolare comunicazione tra client e server. Viene inizializzato a zero ad inizio connessione ed incrementato ad ogni invio di un blocco dati. Il client e il server possiedono numeri di sequenza personali totalmente indipendenti tra di loro.

inizialization_vector(IV): e' un vettore di inizializzazione per l'algoritmo di crittografia simmetrica. L' IV viene sovrascritto con il risultato della cifratura dell' ennesimo blocco, per poi essere utilizzato di nuovo come inizializzatore alla cifratura del blocco successivo.

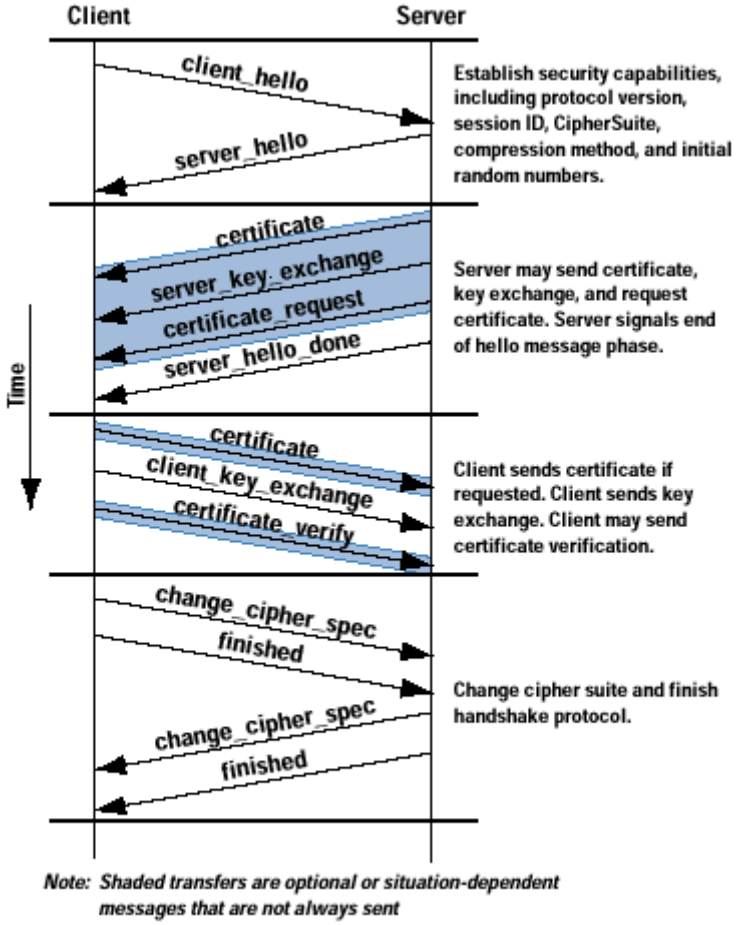
2.PROTOCOLLO

Il protocollo SSL, dovendo essenzialmente garantire un traffico cifrato dei dati tra i due host comunicanti, deve essere in grado di gestire la cifratura e la decifratura dei messaggi che provengono dal protocollo dello strato applicazione. Prima che la comunicazione protetta abbia inizio è però necessaria una breve fase di negoziazione (*handshake*) dei vari parametri di comunicazione. Il protocollo SSL e' stato quindi progettato come somma di due protocolli distinti che assolvono a due precisi compiti: l' *handshake protocol* serve per la negoziazione dei parametri di sessione e connessione; il *record protocol* serve per la gestione dell'effettiva comunicazione cifrata. Nella pila TCP/IP tale suddivisione si traduce in una scissione in due parti del layer SSL, come da figura:



2.1 Handshake protocol

E' la parte più complessa di SSL. Questo protocollo permette al server e al client di autenticarsi e di negoziare chiavi e algoritmi di codifica e di autenticazione per proteggere i dati inviati con il record protocol.



La seguente figura mostra lo scambio di messaggi tra client e server:

Fase 1: è usata per iniziare la connessione e stabilire gli algoritmi di sicurezza da associargli. Lo scambio è iniziato dal client, che invia il messaggio client_hello con i seguenti parametri:

Version: l'ultima versione di SSL conosciuta dal client;
Random: Un numero random generato dal client, formato da 32 bit di timestamp e da 28 bytes generati da un generatore di numeri random. Questo valore è usato per evitare gli attacchi di tipo replay
Session ID: un numero di lunghezza variabile che identifica la sessione. Se diverso da zero indica che il client vuole aggiornare i parametri di una sessione già esistente, zero indica che il client vuole creare una nuova sessione;
ChiperSuite: una lista che elenca gli algoritmi di crittografia conosciuti dal client, in ordine decrescente di preferenza. (RSA o alcuni tipi di Diffie_Hellman)

Compression Method: la lista dei metodi di compressione conosciuti dal client.
Ricevuto il client_hello, il server risponde con Server_hello, un messaggio che contiene gli stessi parametri del client_hello, ma con le seguenti convenzioni:

Version: l'ultima versione di SSL conosciuta dal server;
Random: è generato dal server nello stesso modo del client, i due numeri random sono così indipendenti tra loro;
Session ID: Se il Session ID del client è zero crea un nuovo numero per identificare la nuova sessione, altrimenti restituisce il numero inviato dal client;
ChiperSuite: contiene l'algoritmo di crittografia scelto dal server tra quelli inviategli dal client;
Compression Method: contiene il metodo di compressione scelto dal server tra quelli inviategli dal client.

Fase 2 (server to client): In questa fase i messaggi vengono inviati solo dal server al client:
certificate: il server invia il suo certificato (deve essere autenticato da una CA);
server-key-exchange: dipende dall'algoritmo di cifratura scelto, questo messaggio non è richiesto solo in due casi: se il server sta usando l'algoritmo fixed-Diffie-Hellman o se sta usando RSA-key-exchange.
certificate-request: richiede il certificato del client, questo messaggio non è necessario se il server sta usando l'anonymous-Difffie-Hellman;
server-hello-done: è usato dal server per indicare la fine dei suoi messaggi.

Fase 3 (client to Server): Quando riceve il server_hello_done, il client verifica la validità del certificato del server e se i parametri contenuti nel server_hello sono "accettabili". Se tutti i requisiti sono soddisfatti il client comincia la fase 3, in questa fase l'unico ad inviare messaggi è il client:
certificate: il client invia il proprio certificato (se richiesto dal server);
client-key-exchange: il contenuto del messaggio dipende dall'algoritmo scelto;
certificate-verify: è un messaggio per esplicitare la verifica del proprio certificato.

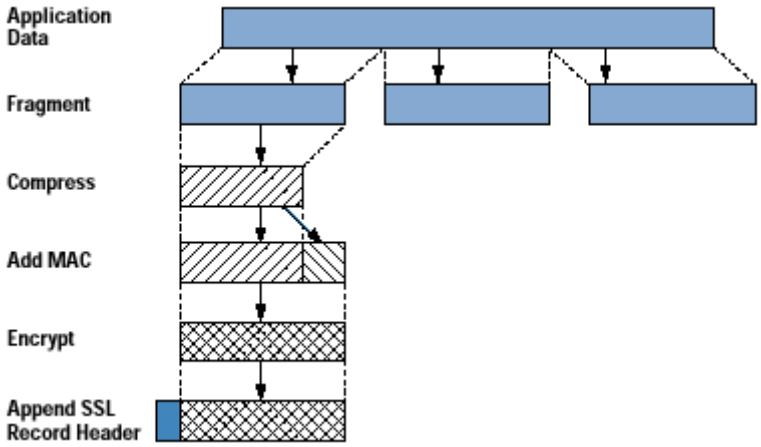
Fase 4: Completa la creazione della connessione sicura, prima il client, poi il server inviano gli stessi due messaggi:
change-cipher-spec: indica gli algoritmi usati per la codifica e quelli da utilizzare per il MAC;
finished: indica la fine del protocollo handshake.

Nell'handshake protocol è previsto anche un protocollo per gli errori; l'**alert protocol**: se ci sono errori viene inviato un messaggio di errore,contentente un intero che indica la gravità dell'errore ("warning"=1, "fatal"=2) e una

stringa che indica il nome del parametro errato. Se l'errore è “fatal” si interrompe subito la sessione, che viene chiusa.

2.2 Record protocol

L'SSL Record Protocol garantisce due goal per le connessioni SSL: la confidenzialità (tramite la codifica dei dati) e l'integrità (tramite il MAC).
La seguente figura illustra le operazioni del protocollo:



Frammentazione: inizialmente il protocollo divide il messaggio in blocchi (di 214 bytes);
Compressione: è opzionale (nell'SSL 3.0 e nel TLS non è più presente)
Aggiunta del MAC: si aggiunge il MAC al messaggio, il MAC è ottenuto tramite le informazioni ricavate dall'handshake;
Codifica: si codifica utilizzando la chiave simmetrica ricavata dall'handshake;
Aggiunta dell'header: si aggiungono alcune informazioni in chiaro al messaggio codificato:
Content Type: indica gli algoritmi usati per creare il “frammento”
Major Version: indica la maggiore versione di SSL in uso (per SSL v3.0 il valore è 3)
Minor Version: indica la minima versione di SSL in uso;
Compressed Length: indica la lunghezza in bits del frammento (o del frammento compresso se si usa una compressione).

3. ANALISI DELLA SICUREZZA

La struttura del protocollo così come è stata presentata sembra prevenire ogni tipo di attacco alla riservatezza e all'integrità dei dati. Questa era anche la convinzione degli sviluppatori di Netscape quando introdussero SSLv3, aggiornando le imperfezioni della versione precedente. Da allora numerosi ricercatori e hacker si sono prodigati nel cercare di sviluppare applicazioni di

attacco, ma i risultati ottenuti sono dovuti soprattutto all'inefficienza di alcune implementazioni, oltre che all'ingenuità degli utenti.
Quelle seguenti sono alcune delle debolezze strutturali trovate finora:

flessibilità: Le cipher suites di SSL supportano vari gradi di sicurezza, per rendere il protocollo adattabile alle diverse esigenze. Questo comporta la possibilità, per l'utente malintenzionato, di costringere i due endpoint a comunicare secondo lo standard di sicurezza più debole. E' quello che accade nel caso di attacchi del tipo *drop change cipher spec message*, e che in qualche modo è legato alla facilità nell' espugnare con forza bruta lo standard di cifratura RC4 . Un gruppo di ricercatori è infatti riuscito a decifrare i messaggi crittografati con RC4 in otto giorni, e questo tempo può essere naturalmente ridotto con l'aumentare della potenza di calcolo dei computer. Come flessibili sono le cipher suites supportate da SSL, altrettanto flessibile è l'algoritmo di negoziazione di tali cipher suites. Il fatto che si possa effettuare una sessione con scambio di chiavi anonimo permette infatti, da una parte, di facilitare interazioni in cui non è necessaria l'autenticazione, ma dall'altra apre la strada ad attacchi del tipo *replay on anonymous key exchange*. Il concetto di flessibilità visto come debolezza può quindi essere concisamente espresso dicendo che SSL è sicuro quanto la più debole cipher suite supportata, poichè un attacco può costringere gli host a comunicare secondo l'algoritmo di cifratura più facilmente decodificabile.

Compatibilità:Si è deciso di rendere SSLv3 compatibile con la versione precedente. Questo permette una minore rigidità nella transizione di aggiornamento, ma allo stesso tempo fornisce un appiglio per gli attacchi di tipo *version rollback*. Se si può costringere due host a comunicare secondo il protocollo SSLv2, sarà molto più facile sfruttare le numerose falle della vecchia versione. Cioè può riuscire a modificare con relativa facilità messaggi critici della fase di handshake, al fine di costringere gli host a comunicare con RC4, lo standard di cifratura più espugnabile.

Denial of Service: Quando si riceve un codice di autenticazione del messaggio (MAC) errato il livello Record di SSL termina la connessione. Questa facilita i cosiddetti attacchi di tipo *Denial of Service*, che non pregiudicano in nessun modo l'integrità o la riservatezza dei dati scambiati, ma semplicemente impediscono al server attaccato di effettuare connessioni di alcun tipo. SSL non è stato sviluppato per difendersi da strategie maliziose di questa categoria, e infatti la sua struttura costituisce addirittura un supporto ad attacchi di servizio negato.

PARTE 2: SETEFI e VirtualCard MONETAONLINE

SETEFI è la società del gruppo bancario Intesa Sanpaolo specializzata nella gestione dei pagamenti con moneta elettronica, ed è diventata leader nel mercato italiano nel settore dell'acquirer, ovvero nel offrire servizi e nel organizzare le operazioni riguardanti richieste di autorizzazione di transazioni di pagamento con carta di credito. In particolare si propone agli esercenti come interlocutore unico per tutte le fasi del processo di incasso tramite POS (point of sale, è uno strumento di incasso che consente il trasferimento di fondi da un soggetto (compratore) ad un altro (esercente venditore di beni o servizi), grazie all'utilizzo di carte di debito (Bancomat), di credito e prepagate). Inoltre SETEFI gestisce direttamente 10 milioni di carte di pagamento emesse dal Gruppo Intesa Sanpaolo assicurandone la personalizzazione, la fase autorizzativa dei pagamenti oltre al regolamento contabile delle transazioni.

1. SETEFI

1.1 ORGANIZZAZIONE

A livello organizzativo, le banche del Gruppo Intesa Sanpaolo emettono carte di pagamento e SETEFI gestisce, in maniera autonoma, i pagamenti con moneta elettronica in qualità di acquirer, gestisce i terminali per quanto riguarda i POS e ha funzione di processor (organizzazione tecnica per quanto le transazioni di pagamento) per le carte di pagamento.

1.2 NORMATIVE

Dal punto di vista normativo, SETEFI ha chiesto ed ottenuto autorizzazione dalla Banca d' Italia per offrire i seguenti servizi di pagamento:

- Esecuzione di ordini di pagamento, anche trasferimento di fondi, su un conto presso un prestatore di servizi di un utilizzatore.
- Esecuzione di pagamenti legati ad una certa linea di pagamento accordata all'utilizzatore di servizi di pagamento.
- Emissione e acquisizione di strumenti di pagamento.

1.3 SERVIZI DI INCASSO

Il servizio che SETEFI offre ai servizi commerciali convenzionati è:

- Garantire agli esercenti convenzionati gli importi relativi alle vendite mediante accettazione delle carte di credito.
- Ricevere le somme che spettano agli operatori commerciali a fronte dell'accettazione delle carte di credito.
- Curare il successivo trasferimento sul conto corrente bancario indicato dall'operatore commerciale.

La commissione corrisposta dall' operatore commerciale costituisce, infatti, la remunerazione del rischio, assunto dall' acquirer (banca che fa operazione), connesso con l' eventuale insolvenza del titolare della carta.

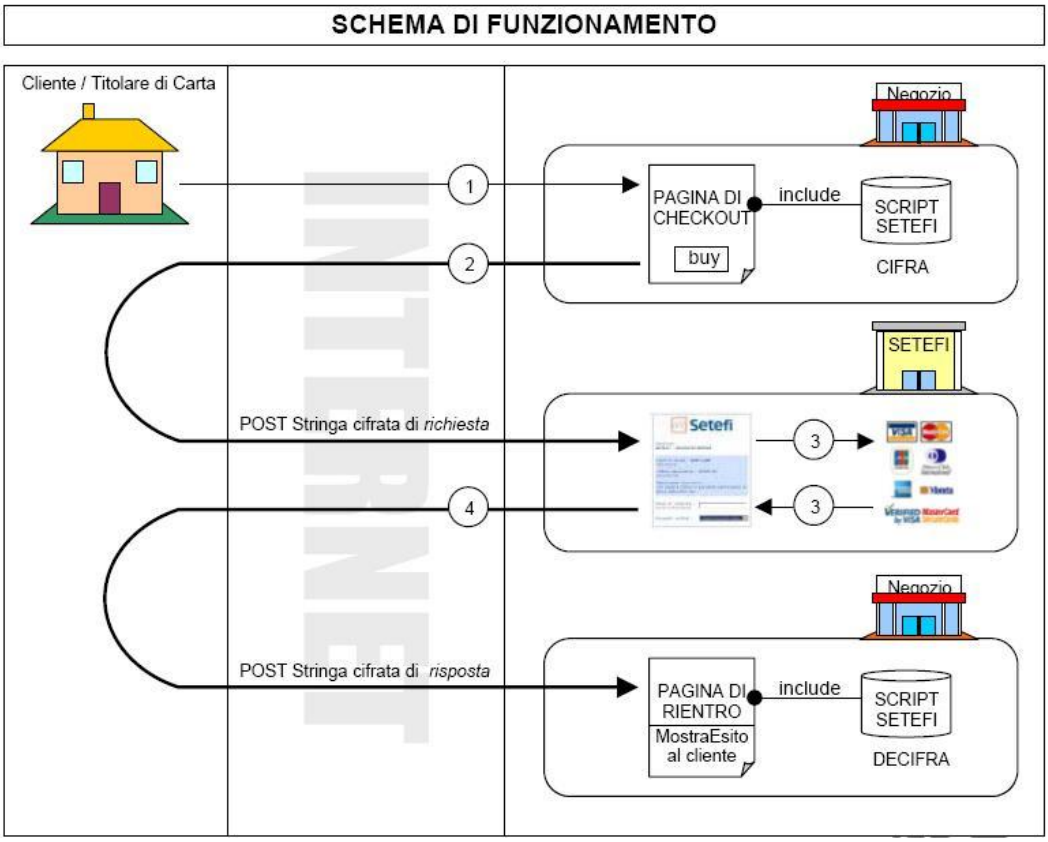
1.4 GESTIONE CARTE DI PAGAMENTO

SETEFI gestisce direttamente gli oltre 10 milioni di carte di pagamento emesse dal Gruppo Intesa Sanpaolo assicurandone:

- Acquisto di plastiche vergini adeguate alle norme previste dai circuiti di pagamento.
- La personalizzazione delle plastiche mediante caricamento dati su microchip e banda magnetica.
- L' invio delle carte di pagamento alle filiali del Gruppo Intesa Sanpaolo o al domicilio del Titolare.
- Autorizzazione dei pagamenti.
- Il regolamento contabile delle transazioni.

1.5 OBIETTIVO PRINCIPALE

Obiettivo principale di SETEFI è quindi garantire ai negozi online la possibilità di accettare il pagamento attraverso carte di credito, lasciando a SETEFI la fase di gestione e acquisizione del codice di carta di credito del cliente finale nonché del processo di autorizzazione.
Per aderire a SETEFI il sito del commerciante deve predisporre due pagine web adibite allo scambio di messaggi con il server SETEFI; una di esse sarà utilizzata per preparare una stringa cifrata contenente i dati della transazione in corso, l'altra per decifrare il messaggio di risposta inviato da SETEFI. Per fare ciò SATEFI invia precedentemente al commerciante un file contenente le informazioni di cifratura e decifratura, che servirà nella operazione detta sopra.



2. VIRTUAL CARD

Per i possessori di una carta di credito emesse dalle banche del gruppo Intesa Sanpaolo è offerto il servizio, attivabile via internet e gestito da SETEFI, VirtualCard MONETAONLINE, che consiste, utilizzando i dati presenti sulle carte "fisiche" del possessore, nella creazione di una Virtual Card da utilizzare per acquisti online. Ci sono due tipologie di carta virtuale:

- Utilizzabile per una singola operazione.
- A tempo.

Per attivare VirtualCard MONETAONLINE serve che la carta "fisica" del possessore abbia un codice utente, e di una password che, a seconda della tipologia di carta che si ha, viene data al momento dell'emissione o spedita a casa.

2.1 COME SI UTILIZZA

Dopo aver acquistato un oggetto via internet viene proposto un modulo d'ordine dove vengono richiesti i dati della propria carta. A questo punto l'utente dà, invece che i dati della sua carta "fisica", quelli della virtual card (numero, scadenza, cvv2, che è un codice di sicurezza di 3 cifre presente sul retro della carta) e quindi continua le operazioni di pagamento come se fosse

la sue vera card, garantendosi così maggior protezione per i propri dati (essendo la card attiva o per una singola transazione o per un tempo limitato).

PARTE 3: ESEMPIO DI APPLICAZIONE

1.FASE CIFRATURA E DECIFRATURA

Le funzioni di cifratura/decifratura fornite da Setefi sono attualmente disponibili per ambienti in grado di supportare le seguenti tecnologie:
- ASP
- PHP
- JAVA

Lo scambio dei messaggi tra il titolare di carta di credito e il pos virtuale Setefi avviene in maniera sicura grazie al protocollo SSL.
Le due funzioni sono:

1. **Rij_Client_CifraNew;**
2. **Rij_Client_DecifraNew;**

1.1 LA FUNZIONE Rij_Client_CifraNew

La prima funzione, viene utilizzata per creare e criptare la stringa contenente i dati della transazione.
E' necessario fornire come parametri della funzione tutti i campi presenti nel tracciato record presente nella pagina successiva.
Ovvero la funzione prenderà in input l'importo senza decimali, il riferimento dell'operazione, data e ora della transazione, il numero dell'operazione, la descrizione e il codice nazione più alcuni campi chiamati Filler che sono insieme di bit di riempimento usati tipicamente nell'impacchettamento dei messaggi per raggiungere un numero di bit, o un multiplo di esso, prefissato.

CAMPO	LUNGHEZZA MAX	VALORE DA IMPOSTARE	DESCRIZIONE	ESEMPIO
purchase_amount	12		Importo comprende anche i due decimali senza segni di punteggiatura	19,80 * 100 = 1980
Filler	1	"2"	Fisso a 2	
Filler	3	"978"	Fisso a 978	
rifOperazione	18		Riferimento operazione commerciante. **Deve essere univoco in assoluto.** E' anche, di norma, riportato nell'estratto conto del cliente	E' il riferimento dell'operazione noto anche al cliente finale. Esempio: numero fattura, codice operazione Internet, etc.
DataOperazione	6		Data operazione AAMMGG	030423 = 23 Aprile 2003
OraOperazione	6		Ora operazione HHMMSS	183000 = 18:30.00
NumOperazione	4		Numero progressivo operazione, gestito dal commerciante. Deve essere univoco nella giornata.	
Descrizione	64		Descrizione merce/servizio	Tenda igloo
Filler	19	""	Filler	
Filler	4	""	Filler	
Filler	2	""	Filler	
Filler	8	""	Filler	
Filler	2	""	Filler	
Filler	1	""	Filler	
Filler	3	""	Filler	
Filler	32	""	Filler	
Filler	28	""	Filler	
CodNazione	3	"380"	Fisso a 380	
Filler	15	""		
Filler	1	""		
Filler	3	""		
Filler	58	""		

La funzione restituisce in risposta una stringa cifrata, che dovrà essere inviata

tramite post http al server Setefi , come si evince dall’esempio seguente:

CHIAMATA ALLA FUNZIONE "JAVASCRIPT" Rij_Client_CifraNew INCLUSA NEL FILE 55555555.asp , in una pagina ASP

```
<!-- #include FILE = ".\55555555.mph" -->
<%
Dim MPIUrl
Dim purchase_amount,nDecimali,codDivisa,rifOperazione,DataOperazione,OraOperazione,NumOperazione,Descrizione,codNazione
purchase_amount = 190*100 ' L = 12
rifOperazione = "123456789 12345678"
dataOperazione = 030409
oraOperazione = 135040
numOperazione = "1234"
descrizione = "Computer palmare Dell"
DataMerchCifrati = Rij_Client_CifraNew(purchase_amount,”2”,”978”,rifOperazione,DataOperazione,OraOperazione,NumOperazione,Descrizione,””,””,””,””,””,””,””,””,””,””,”380”,””,””,””,””,””)
%>
<html>
<head>
<title>Negozio Demo</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" leftmargin="0" topmargin="0">
<form name="PaymentRequestForm" action=” https://www.monetaonline.it/MPI/MPIRequest.asp " method="POST">
<input type="hidden" name="PaymentRequest" value="<%= DataMerchCifrati %>">
<input type="submit" name="Submit" value="buy now">
</form>
</body>
</html>
```

CODICE ELABORATO

```
<html>
<head>
<title>Negozio Demo</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" leftmargin="0" topmargin="0">
<form name="PaymentRequestForm" action="https://www.monetaonline.it/MPI/MPIRequest.asp" method="POST">
<input type="hidden" name="PaymentRequest" value="1154c0b04a698fe0c6bbc0f6525b2e690b39ba4b3da7fd2de4310c12d6cfcdea3e7c7fec1a21c94c7e78712c27e986884fe84c38176a177a7e8fed4e1f75b3e2b06b941cad12e53346ee1c9746596eda828558959faddc34093f21be7b81a68615c8eeea5bee37f0b5c0fd92d945ce9ba933b3a665a12434e68d20a1e42fe9e4784b76c3cb0f6fef5501482bee15da389d6826e532da54133d0dd1eed1af5a375d537c735a3eede2127c0f41b9e9df0b10a918f8e80ae2e41b5a5981549da4df2537b638414f20b6a18175af2fc707c0eb6d37d28474890f017cc7c0cc947560fb3e613da4a70e2944e405063b022497012174.key0094>
<input type="submit" name="Submit" value="buy now">
</form>
</body>
</html>
```

Abbiamo evidenziato i tratti salienti dell’esempio:

- La chiamata alla funzione in questione, a cui passiamo i parametri visti prima nella tabella;
- Il form in cui c'è un input di tipo "hidden", ovvero nascosto al client, valorizzato con la stringa cifrata;

Le ultime righe evidenziate sono un esempio in codice elaborato.

1.2 LA FUNZIONE Rij_Client_DecifraNew

Viene utilizzata per ottenere la decifratura del messaggio di risposta prodotto da Setefi.
E' necessario fornire come parametro della funzione, il campo ricevuto dal post http inviato da Setefi tramite il browser. Ecco un esempio:

CHIAMATA ALLA FUNZIONE "JAVASCRIPT" Rij_Client_DecifraNew INCLUSA NEL FILE 55555555.asp, in una pagina ASP

```
<%response.buffer=true
Response.Expires=-1%>
<!-- #include FILE = ".\55555555.mph" -->
<%
dim campoDaDecifrare, stDecifrata, risposta
campoDaDecifrare = request.form("PaymentResponse")
if campoDaDecifrare <> "" then
strDecifrata = Rij_Client_DecifraNew(campoDaDecifrare)
if left(strDecifrata,3)="000" then
'Se I primi 3 caratteri del msg decifrato sono uguali a "000" allora la transazione è andata a buon fine
risposta = "LA TRANSAZIONE E' ANDATA A BUON FINE"
else
risposta = "ATTENZIONE LA TRANSAZIONE NON E' ANDATA A BUON FINE"
end if
else
response.write "ERRORE, campo da decifrare VUOTO"
end if
%>
<html>
<head>
<title>Negozio Demo</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" leftmargin="0" topmargin="0">
<%
response.write risposta
%>
</body>
</html>
```

CODICE ELABORATO

```
<html>
<head>
<title>Negozio Demo</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" leftmargin="0" topmargin="0">
LA TRANSAZIONE E' ANDATA A BUON FINE / ATTENZIONE LA TRANSAZIONE NON E' ANDATA A BUON FINE
</body>
</html>
```

Come detto sopra, e come si evidenziato nell'esempio, alla funzione di decifratura bisogna dare in input il campo ricevuto dal post http inviato da

Setefi tramite il browser.
Ovviamente questa risposta di Setefi verso il commerciante segue delle regole descritte nel tracciato record sotto riportato:

CAMPO	POSIZIONE	LUNGHEZZA	DESCRIZIONE
statoTrans	1	3	Se = 000 allora la transazione è andata a buon fine
Data	4	8	Formato AAAAMMGG
Ora	12	6	Formato HHMMSS
Filler	18	6	
Cod. Autoriz.	24	6	Se autorizzata è valorizzato ed è diverso da "000000"
Descrizione esito	30	29	Descrive lo stato dell'esito eventualmente da presentare al cliente
Rif. Setefi	59	12	Codice di riferimento attribuito da Setefi alla transazione.
Filler	71	4	
Tipo pagamento	75	1	Modalità: 5 = 3Dsecure 6 = 3Dsecure 7 = Normale
Filler	76	8	
Purchase_amount	84	12	Importo – uguale a quello impostato in fase di richiesta dal commerciante
Filler	96	4	
RifOperazione	100	18	Riferimento operazione – uguale a quello impostato in fase di richiesta dal commerciante
DataOperazione	118	6	Data operazione AAMMGG – uguale a quella impostata in fase di richiesta dal commerciante
OraOperazione	124	6	Ora operazione HHMMSS – uguale a quella impostata in fase di richiesta dal commerciante
NumOperazione	130	4	Numero progressivo operazione – uguale a quello impostato in fase di richiesta dal commerciante
Descrizione	134	64	Descrizione merce/servizio – uguale a quella impostata in fase di richiesta dal commerciante
Filler	198	179	
Messaggio	377	80	Messaggio di risposta eventualmente da presentare al cliente

Ovvero lo stato della transazione, la data e l'ora, il codice di autorizzazione, la descrizione dell'esito, il riferimento di Setefi, il tipo di pagamento, l'importo, il riferimento, la data e l'ora dell'operazione, il numero dell'operazione la descrizione e un messaggio. Anche in questo caso ci saranno i bit di riempimento chiamati Filler.

2. FASE DI RISPOSTA

In questa fase è trattata la gestione dell'URL di risposta.
Per URL di "risposta" si intende l'indirizzo Internet abilitato dal commerciante

alla ricezione del messaggio di esito elaborato da Setefi.
La URL può essere comunicata a Setefi oppure gestita dinamicamente per ogni richiesta di pagamento.
La gestione dinamica prevede l'inserimento di un campo "hidden" nella form contenente la stringa cifrata da inviare a Setefi. Il nome da assegnare al predetto campo deve essere "RETURL" (come si vede nell'esempio seguente) e il suo contenuto deve corrispondere alla URL dalla quale il commerciante desidera interpretare il messaggio di risposta Setefi.

Es:
<form name="PaymentRequestForm" action="https://www.monetaonline.it/MPI/MPIRequest.asp" method="POST">
<input type="hidden" name="PaymentRequest" value="F383298A89239C893932832….."> <input type="hidden" name="RETURL" value=http://www.sitopreferito.it/risposta.php>
<input tpe="submit" name="Submit" value="buy now">
y</form>

3. CONTABILIZZAZIONE

Il commerciante può in funzione delle proprie esigenze, avvalersi di uno dei tre differenti metodi di contabilizzazione delle transazioni autorizzate.
I primi due metodi utilizzano le funzioni di contabilizzazione presenti sul sito www.monetaonline.it e sono:

- Il primo metodo, implicito, prevede la contabilizzazione automatica a fine giornata di tutte le richieste autorizzate nella stessa giornata e non cancellate dal commerciante tramite l'apposita funzione presente sul sito www.monetaonline.it.

- Il secondo metodo, esplicito, richiede che il commerciante effettui personalmente la contabilizzazione delle operazioni autorizzate tramite l'apposita funzione presente sul sito www.monetaonline.it. Il commerciante può confermare operazioni autorizzate anche nelle precedenti giornate, purché non anteriori a 20 giorni di calendario.

Il terzo metodo, esplicito a mezzo archivio elettronico, richiede la creazione di tale archivio a cura del commerciante. L'archivio deve contenere tutte e sole le operazioni autorizzate che si intendono contabilizzare, e può essere successivamente inviato a SETEFI tramite la funzione "INVIA FILE" presente nel sito www.monetaonline.it.

3.1 METODO IMPLICITO

E' il metodo scelto dal commerciante che intende procedere alla contabilizzazione automatica a fine giornata delle transazioni autorizzate (es.: commerciante che vende beni digitali o servizi).
Il commerciante può richiedere nel corso della giornata l'eliminazione di una o più operazioni precedentemente autorizzate, utilizzando l'apposita funzione "VARIA STATO OPERAZIONE" dell'Area Operatori Commerciali disponibile sul

sito Internet di SETEFI all'indirizzo http://www.monetaonline.it/O_default.asp fornendo i dati richiesti.
Tali dati possono essere rilevati utilizzando l'apposita funzione "ELENCO AUTORIZZAZIONI RICHIESTE SU INTERNET" dell'Area Operatori Commerciali disponibile sul sito Internet di SETEFI di predetto indirizzo.

3.2 METODO ESPLICITO

E' il metodo scelto dal commerciante che **non** intende procedere alla contabilizzazione automatica delle transazioni autorizzate. E', ad esempio, il caso del commerciante che vende beni fisici e che procederà alla contabilizzazione delle transazioni autorizzate solo dopo la spedizione del bene.
Il commerciante richiede la contabilizzazione, utilizzando l'apposita funzione "VARIA STATO OPERAZIONE" dell'Area Operatori Commerciali disponibile sul sito Internet di SETEFI all'indirizzo:
http://www.monetaonline.it/O_default.asp
fornendo i dati richiesti. E' possibile richiedere la contabilizzazione di un importo minore o uguale all'importo autorizzato.
I dati richiesti per la conferma possono essere rilevati utilizzando l'apposita funzione "ELENCO AUTORIZZAZIONI RICHIESTE SU INTERNET" dell'Area Operatori Commerciali disponibile sul sito Internet di SETEFI di predetto indirizzo.

BIBLIOGRAFIA

1.**The Internet Protocol Journal - Volume 1, No. 1 - Cisco Systems** (www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_1-1/ssl.html)

2.**gianluca82.altervista.org/ssl/ssl3.html**

3. **Appunti di sicurezza delle reti, Giorgio Faina**

4. **www.setefi.it**

5. **www.monetaonline.it**

6. **Documento Setefi MPI_Standard.doc/SC_20051214**