MINISTERO DELLO SVILUPPO ECONOMICO
DIREZIONE GENERALE PER LA LOTTA ALLA CONTRAFFAZIONE
UFFICIO ITALIANO BREVETTI E MARCHI

DOMANDA DI INVENZIONE NUMERO	**102011901915490**
Data Deposito	**11/02/2011**
Data Pubblicazione	**11/08/2012**

Classifiche IPC

Titolo

METODO PER LO SCAMBIO DI DATI SICURO NELLE COMUNICAZIONI SIMMETRICHE.

Metodo per lo scambio di dati sicuro nelle comunicazioni simmetriche

---

La presente invenzione riguarda un metodo per lo scambio di dati sicuro nelle comunicazioni simmetriche.

Più precisamente, la presente invenzione riguarda un metodo per le comunicazioni simmetriche, in particolare per crittare e decrittare messaggi con lunghezza non stabilita a priori. Perciò il metodo permette la comunicazione crittata di flussi di dati (stream) generati in tempo reale, ovvero di dati la cui lunghezza finale non è predicibile.

La denominazione comune QP-DYN si riferisce ad una famiglia di algoritmi crittografici a flusso, basati su una struttura matematica comune, ma che differiscono nei protocolli specifici.

La limpidità della loro struttura matematica permette di introdurre una molteplicità praticamente infinita di varianti, mantenendo le caratteristiche di robustezza.

Gli algoritmi QP-DYN sono simmetrici, ma la chiave privata è composta di due parti, una fissa (parte simmetrica), che ha il ruolo di generatore di chiavi segrete condivise (SSK nel seguito) e una che può essere resa pubblica (parte asimmetrica).

I due interlocutori possono cambiare arbitrariamente e a costo zero la parte potenzialmente pubblica della chiave ottenendo ogni volta delle chiavi segrete condivise (SSK) differenti.

Ciò esclude gli attacchi per accumulazione di informazioni attraverso l'uso ripetuto della stessa chiave.

Come tutti gli algoritmi crittografici a flusso anche i protocolli QP-DYN sono costituiti da tre fasi indipendenti:

(I) Algoritmo di inizializzazione;

(II) Algoritmo per la produzione della chiave segreta condivisa (SSK) data l'inizializzazione;

(III) Algoritmo di codifica e decodifica, cioè uso della SSK per scambiare messaggi.

Gli algoritmi della classe (III) possono essere qualsiasi tuttavia, dato che nel caso dei metodi QP-DYN la SSK è lunga quanto il testo ed ha ottime proprietà statistiche, mentre la parte potenzialmente pubblica della chiave può essere cambiata per ogni testo a costo zero, come algoritmo di codifica e decodifica è sufficiente usare lo XOR in modalità "*one time pad*" che, per il teorema di Shannon, è quella di massima sicurezza.

Il rapporto [Gabler07], basato su una parte della tesi di Ph.D. di Markus Gäbler (Cottbus), discute i risultati di un'analisi statistica dei generatori QP-DYN di sequenze pseudo-casuali.

Questa analisi è stata ripetuto ormai molte volte dal gruppo di ricerca della Richiedente confermando i risultati ottenuti.

Il rapporto [ItaOttGrilLent09] è stato realizzato dal gruppo di ricerca diretto dal prof. Giuseppe Italiano del Dipartimento di Informatica, Sistemi e

Produzione, dell'Università di Roma Tor Vergata e confronta le prestazioni della esecuzione su telefoni cellulari i metodi QP-DYN con parecchie varianti di suite di metodi crittografici attualmente più usate sia a livello di crittografia a chiave pubblica (RSA, Diffie-Hellmann, curve ellittiche) che di flusso (AES, RC5).

Il risultato è che i metodi sviluppati dalla Richiedente sono teoricamente più robusti e producono SSK più lunghe in tempi più brevi.

Cenni sulle basi teoriche degli algoritmi QP-DYN

Le basi teoriche degli algoritmi QP-DYN sono nella teoria dei sistemi dinamici caotici e sono descritte nel lavoro [AbAcAu91] che sviluppa il lavoro precedente [AcdeTiDiLi81], nel quale viene introdotta l'idea principale del metodo, aggiungendo una dimostrazione costruttiva del fatto che, scegliendo opportunamente il dato iniziale, è possibile ottenere periodi arbitrariamente lunghi e fornisce una stima della lunghezza del periodo in funzione del dato iniziale. Questi lavori contengono anche (sia pure in forma implicita) la spiegazione del perché il passaggio dal metodo di generazione di vettori pseudo-casuali all'algoritmo crittografico sia tutt'altro che banale: il fatto è che, mentre i computer possono lavorare solo su numeri razionali, la quasi totalità dei risultati sui sistemi dinamici caotici (in particolare la dimostrazione delle loro proprietà di caoticità) usano

tecniche di teoria della misura e quindi escludono insiemi di misura nulla quali i numeri razionali.

A questo c'è da aggiungere che una pedissequa implementazione software del sistema dinamico alla base dell'algoritmo darebbe luogo ad un protocollo crittografico facilmente rompibile (per maggiori dettagli si veda la sezione 11). Quindi la simulazione del sistema dinamico caotico deve essere integrata con accorgimenti di natura specificamente crittografica.

Dopo la proposta originaria contenuta nel lavoro [AcdeTiDiLi81], altri autori (alcuni anche recentemente) hanno sviluppato l'idea di utilizzare gli automorfismi iperbolici del toro (o loro varianti) per la generazione di vettori casuali, tuttavia alla Richiedente non risulta la presenza in letteratura di un'applicazione crittografica di tali metodi e algoritmi. Il fine di ogni algoritmo della famiglia QP-DYN è il seguente: "dato in input un testo T di lunghezza binaria $l_T \in \mathbb{N}$, produrre una chiave di lunghezza binaria uguale ad l_T".

Tale metodo viene usato in due situazioni concettualmente diverse:

(i) la lunghezza del testo è nota a priori; oppure

(ii) la lunghezza del testo non è nota a priori.

Il primo caso si presenta tipicamente nell'archiviazione di dati, il secondo nella crittazione di flussi di informazioni.

3. Sistemi dinamici alla base degli algoritmi QP-DYN

Gli algoritmi crittografici QP-DYN vengono realizzati utilizzando varianti della classe dei sistemi dinamici caotici a tempo discreto descritta nella presente sezione, che sono determinati da:

(i1) un intero naturale $d \in \mathbb{N}$, ove $\mathbb{N}$ è l'insieme dei numeri naturali positivi, denominato "dimensione dell'algoritmo";

(i2) una matrice quadrata $\mathbf{M}$ a coefficienti interi naturali e di dimensione $d \times d$ (scriveremo nel seguito sinteticamente $M \in M(\mathrm{d};\mathbb{N})$), che rappresenta la legge del moto del sistema dinamico e che viene scelta invertibile (sistema dinamico reversibile);

(i3) un intero naturale $p \in \mathbb{N}$, detto "modulo".

Quindi un tale sistema dinamico è determinato essenzialmente da 3 oggetti:

$$\{d, M, p\} \qquad (1)$$

p è tipicamente è un numero primo che, per cautelarsi contro ricerche esaustive, conviene sia grande. Se p è mantenuto segreto, gli attacchi che cercano di rompere il metodo crittografico sono molto più difficili. Inoltre, se p non è primo, allora il campo $\mathbb{Z}_p$ dei numeri a $\mathrm{modulo}\ p$ non è più un campo ma solo un anello e non tutti i numeri ($\mathrm{modulo}\ p$) diversi da zero sono invertibili, ciò che rende più difficile il calcolo e la costruzione di matrici invertibili.

Nel seguito, ci si servirà della seguente:

Definizione 1. *Dato un vettore v_0 a d componenti in $\mathbb{N}$ (sinteticamente, $v_0 \in \mathbb{N}^d$), chiamato "vettore iniziale", si definisce orbita del sistema dinamico $\{d, M, p\}M$ con vettore iniziale v_0 l'insieme:*

$$O(M, v_0) = \left\{ M^n v_0 \,(\mathrm{mod}\, p) \in Z_p{}^d \,\middle|\, n \in \right\}$$

dove l'indice $n \in \mathbb{N}$ viene interpretato come parametro temporale discreto: ogni unità di tempo etichetta un passo dell'algoritmo crittografico associato al sistema dinamico, e dove $\mathbb{Z}_p{}^d$ è lo spazio dei vettori di dimensione d a componenti nei numeri interi modulo p.

Poiché ci sono esattamente p^d vettori differenti in $\mathbb{Z}_p{}^d$, ogni orbita $O(M, v_0)$ è un insieme finito. Inoltre, se

$$\det(M) \neq 0 \ (\mathrm{mod}\, p)$$

allora per ogni vettore v_0 esiste un $P \in \mathbb{N}$ tale che $v_T = v_0$.

Se P è il più piccolo di tali numeri, si dice che l'orbita ha periodo P.

Poiché il sistema dinamico è deterministico e reversibile, un'orbita può intersecare se stessa soltanto se ritorna al vettore iniziale v_0 ed in questo caso si ripete.

Scopo della presente invenzione è quello di fornire un metodo di crittografia che risolva i

problemi e superi gli inconvenienti della tecnica anteriore.

E' ulteriore scopo specifico della presente invenzione quello di fornire i mezzi e gli apparati per l'implementazione del metodo scopo dell'invenzione.

E' oggetto della presente invenzione un metodo di crittografia per lo scambio di dati tra due interlocutori, che utilizza i vettori di un sistema dinamico a tempo discreto determinato da:

(i1) un intero naturale $d \in \mathbb{N}$, ove $\mathbb{N}$ è l'insieme dei numeri naturali positivi, chiamato "dimensione dell'algoritmo";

(i2) una matrice quadrata $\mathbf{M}$ a coefficienti interi naturali e di dimensione $d \times d$, che rappresenta la legge del moto del sistema dinamico;

(i3) un intero naturale $p \in \mathbb{N}$, chiamato "modulo", preferibilmente un numero primo;

(i3) un intero naturale $n \in \mathbb{N}$, che rappresenta il contatore del passo n-esimo di iterazione della legge del moto;

(i4) un vettore iniziale v_0, ed un vettore $v_n = M^n v_0$ relativo al passo di iterazione n-esimo,

per generare una chiave binaria di crittazione k, definendo, per ogni $n \in \mathbb{N}$, una funzione $k_n : \mathbb{N}^{dn} \to \mathbb{N}$, denominata "funzione generatrice della chiave di ordine n" o "KGF" che usa l'espansione binaria delle componenti degli n vettori prodotti nei primi n passi di interazione della legge del moto per costruire la chiave parziale n-esima, il metodo comprendendo

l'esecuzione delle seguenti fasi successive:

A. cominciando da v_0 al primo passo 0, calcolare n passi di iterazione della legge del moto calcolando sempre le componenti dei vettori modulo p, avendo così all'n-esimo i vettori $\{v_0, v_1, \ldots, v_n\}$;

B. calcolare il vettore al passo $(n+1)$-esimo:

$$v_{n+1} := Mv_n \quad (\text{modulo } p)$$

C. calcolare la chiave parziale $(n+1)$-esima:

$$k_{n+1}(v_1, \ldots, v_{n+1})$$

D. procedere al calcolo della funzione KGF al passo successivo ripetendo le fasi B e C, fino al passo $n = H \in \mathbb{N}$ in cui la chiave crittografica $k_H = k$ ha la lunghezza desiderata;

il metodo essendo caratterizzato dal fatto che:

I. nella fase C, data una funzione:

$$\lambda : \{\textit{interi naturali di b bits}\} \to \bigcup_{m=1}^{b} \{0,1\}^m$$

con $b \in \mathbb{N}$ e che associa ad un intero naturale di b bit una sotto-stringa della sua espansione binaria, la KGF è la funzione $k_\lambda : \mathbb{N}^d \times \mathbb{N} \to \mathbb{N}$ e $k_{n+1} = k_\lambda(n_1, \ldots, n_d; k_n)$, $n_1, \ldots, n_d \in \mathbb{N}$, $n_1, \ldots, n_d$ essendo le componenti del vettore v_{n+1};

II. si sceglie un numero $q \in \mathbb{N}$ tale che:

$$q \geq Log(p)$$

e per ogni componente di vettore, nelle fasi A e B, si effettua un'operazione di modulo q prima della suddetta operazione modulo p, preferibilmente con $q = 2^{32}$;

III. nella fase B:

- si confronta il vettore corrente v_{n+1} con il vettore iniziale v_0;
- se $v_{n+1} \neq v_0$, si passa direttamente alla fase C;
- se $v_{n+1} = v_0$, si ridefinisce:

$$v_{n+1} := J(v_n)$$

dove $J : \mathbb{N}^d \to \mathbb{N}^d$ è una funzione di salto d'orbita, e si passa alla fase C, l'operazione essendo equivalente a ricominciare una nuova orbita dal vettore $J(v_n)$, che quindi diventa il nuovo punto iniziale;

IV. si effettuano:

- le fasi da A a C per almeno due sistemi dinamici definiti da corrispondenti differenti insiemi $\{d, M, p, k_n, J\}$ e $\{d', M', p', k'_n, J'\}$ che danno luogo ad almeno due rispettive chiavi parziali

$k_{n+1}(v_1,\ldots,v_{n+1})$ e $k'_{n+1}(v'_1,\ldots,v'_{n+1})$; e

- alla fine della fase C la chiave parziale $(n+1)$-esima è la chiave parziale combinata $\overline{k_{n+1}}(v_1,\ldots,v_{n+1};v_1',\ldots,v_{n+1}')$ ottenuta effettuando l'operazione di XOR tra dette almeno due chiavi parziali, dove, se dette almeno due chiavi parziali sono di lunghezza differente, si rendono della stessa lunghezza aggiungendo o eliminando cifre, ed n può essere arbitrario oppure $n=H$.

Preferibilmente secondo l'invenzione, le matrici M e M' di detti almeno due sistemi dinamici sono invertibili e selezionano pertanto corrispondenti sistemi dinamici reversibili.

Preferibilmente secondo l'invenzione, la KGF è la funzione di accodamento sinistro:

$$k_\lambda : \mathbb{N}^d \times \mathbb{N} \to \mathbb{N}$$

definita da:

$$k_{n+1} = k_\lambda(n_1,\ldots,n_d;k_n) := \left[\lambda(n_d),\ldots\lambda(n_1),s\right] \quad ; \quad n_1,\ldots,n_d,s \in \mathbb{N}$$

dove il membro più a destra denota la stringa binaria ottenuta accostando nel modo indicato le stringhe binarie $\lambda(n_d),\ldots,\lambda(n_1),n$, $n_1,\ldots,n_d$ essendo le componenti del vettore v_{n+1}.

Preferibilmente secondo l'invenzione, nell'operazione I detta sotto-stringa risulta da un

taglio di un numero di cifre, partendo dai bit di potenza maggiore, pari ad un numero intero $\tau_{n,v_{nj}}$ variabile da 0 a q e che dipende da n e dalla j-esima componente del vettore v_n, ottenendo così una chiave k con $\sum_{n=1}^{N}\sum_{j=1}^{d}(q-\tau_{n,v_{nj}})$ bit.

Preferibilmente secondo l'invenzione, nell'operazione I detta sotto-stringa risulta da un troncamento deterministico di ordine τ, τ essendo un numero intero positivo, che consiste nel taglio dei primi $\tau_{n,v_{nj}} = \tau$ bit di ogni componente del vettore v_n partendo da sinistra o da destra.

Preferibilmente secondo l'invenzione, nell'operazione I detta sotto-stringa risulta dal taglio dei primi zeri da sinistra fino al primo "1" incluso di ogni componente del vettore v_{n+1} partendo da sinistra o da destra, e/o, alla fine dell'operazione IV, la chiave parziale combinata risulta da detta operazione di XOR e dal successivo taglio dei primi zeri da sinistra fino al primo "1" incluso partendo da sinistra o da destra.

Preferibilmente secondo l'invenzione, la funzione $J:\mathbb{N}^d \to \mathbb{N}^d$ è definita da:

$$J:\mathrm{F}^d \ni v \to \begin{pmatrix} 20\cdots0 \\ 01\cdots0 \\ \vdots\ddots\vdots \\ 00\cdots1 \end{pmatrix} v + \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{N}^d$$

ovvero, esplicitamente:

$$J\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{pmatrix} := \begin{pmatrix} 2v_1 \\ v_2 + 1 \\ \vdots \\ v_d + 1 \end{pmatrix},$$

dove l'uguaglianza è intesa modulo p.

Preferibilmente secondo l'invenzione, nell'operazione IV il campo ambiente è duplicato, ovvero $d = d'$, $M = M'$ e $p \neq p'$, ed inoltre $J = J$ e $k_n = k'_n$.

Preferibilmente secondo l'invenzione, la chiave privata ha una parte potenzialmente pubblica ed una segreta, caratterizzato dal fatto che la parte potenzialmente pubblica è costituita dal vettore iniziale v_0, e la parte segreta, anche detta parte simmetrica, è costituita dalla matrice M, mentre la SSK è la chiave crittografica k, detti due interlocutori cambiando arbitrariamente la parte potenzialmente pubblica della chiave privata al fine di ottenere, per differenti comunicazioni di dati, di volta in volta delle chiavi segrete condivise SSK differenti.

Preferibilmente secondo l'invenzione, il metodo comprende una procedura di inizializzazione, in cui il vettore iniziale v_0 è arbitrariamente definito, attraverso una funzione arbitraria $f(v_0)$, che è tenuta segreta, e/o si scartano i primi $h \in \mathbb{N}$ passi dell'orbita.

E' ulteriore oggetto specifico della presente invenzione un programma per elaboratore caratterizzato

dal fatto di comprendere mezzi a codice atti ad eseguire, quando operano su un elaboratore, il metodo oggetto dell'invenzione.

E' ulteriore oggetto specifico della presente invenzione un supporto di memoria leggibile da un elaboratore, avente un programma memorizzato su di esso, caratterizzato dal fatto che il programma è il programma oggetto dell'invenzione.

L'invenzione verrà ora descritta a titolo illustrativo ma non limitativo, con particolare riferimento alle sue preferite forme realizzative.

4. Uso di una molteplicità di sistemi dinamici

La robustezza crittografica della SSK, costruita nelle sezioni seguenti, si basa sul fatto che, nei casi generici, la ricostruzione della legge dinamica di un sistema dinamico (nel nostro caso la chiave segreta M), a partire dalle informazioni parziali sulle relative orbite, è un problema molto difficile anche se il sistema produce orbite relativamente semplici.

Per esempio, risalire alla legge gravitazionale di Newton dalle orbite ellittiche del sistema solare (Kepler) ha richiesto quasi un secolo di duro lavoro dei migliori matematici, fisici ed astronomi di quel tempo. Per i sistemi caotici il problema è notoriamente ancor più difficile. La difficoltà è enormemente aumentata se le informazioni sono espresse in funzione delle orbite di due o più sistemi dinamici. Questa osservazione intuitiva è alla base della costruzione descritta nella sezione 10. Il sistema dinamico sopra

descritto ha discrete proprietà caotiche ma, come descritto nelle prime righe della sezione 11, dove viene descritto un attacco che permette di ricostruire esattamente la matrice che genera la sequenza pseudo – casuale, esso è facilmente attaccabile dal punto di vista crittografico.

Ciò dimostra che il meccanismo di generazione pseudocasuale non può essere usato da solo come algoritmo crittografico.

Al fine di ottenere un algoritmo crittografico robusto sono state introdotte le seguenti ulteriori operazioni il cui ruolo è quello di distruggere le informazioni algebriche presenti nel meccanismo di generazione pseudo-casuale:

(i) il taglio di bit (si veda la sezione 9.1);
(ii) lo XOR con un'altra sequenza prodotta in modo analogo (si veda la sezione 9);
(iii) il troncamento macchina (si veda la sezione 7);
(iv) il salto d'orbita (si veda la sezione 6.1).

La sezione 10 è dedicata a stimare precisamente quanta parte della struttura algebrica può essere recuperata dopo la sola operazione di taglio e a quale costo computazionale. La stima viene qui fatta per semplicità nel caso di taglio fisso. Nel caso di taglio casuale, la complessità cresce.

5. Produzione della chiave segreta condivisa (SSK)

Il sistema dinamico viene usato secondo l'invenzione per produrre una chiave segreta condivisa (SSK) nel modo descritto nel seguito. Ogni passo del

metodo secondo l'invenzione produce un vettore d-dimensionale a componenti intere (più precisamente nell'insieme $\{0,1,\ldots,p-1\}$).

Ciascuna di queste componenti viene rappresentata in base 2 con c cifre binarie (tipicamente $c=32$). Quindi ogni passo dell'algoritmo produce una stringa di $d \cdot c$ bit. Di conseguenza, dopo n passi dell'algoritmo si avrà una stringa di $n \cdot d \cdot c$ bit. Tale stringa viene usata per costruire una SSK.

Identificando un intero naturale alla sua espansione binaria, tale costruzione si realizza costruendo, per ogni $n \in \mathbb{N}$, una funzione $k_n : \mathbb{N}^{dn} \to \mathbb{N}$, detta "funzione generatrice della chiave (KGF) di ordine n" che usa gli n vettori prodotti nei primi n passi dell'algoritmo per costruire la chiave parziale n-esima (si veda la sezione 9.1 per la scelta effettuata concretamente in una forma realizzativa dell'invenzione).

6. Funzione di salto di orbita

Si è visto che, dato che si lavora modulo p, lo spazio dei possibili vettori, e a fortiori ogni singola orbita di ogni sistema dinamico il cui spazio degli stati è uno spazio vettoriale di dimensione finita sul campo $\mathbb{Z}_p$, contiene un numero finito di punti. Per ottenere buone proprietà statistiche, condizione desiderabile per una buona sequenza crittografica, occorre fare in modo che tali orbite siano molto lunghe.

A tal fine l'originario sistema dinamico viene modificato come segue. L'algoritmo confronta, ad ogni passo, il vettore corrente v_n con il vettore iniziale (è inutile memorizzare tutta l'orbita poiché il sistema è reversibile). Se i due coincidono, sostituisce v_n con $J(v_{n-1})$ dove $J:\mathbb{N}^d \to \mathbb{N}^d$ è una funzione, detta "funzione di salto di orbita".

Ciò è equivalente a ricominciare una nuova orbita dal vettore $J(v_{n-1})$, che quindi diventa il nuovo punto iniziale. È chiaro dalla descrizione qui sopra che il ruolo della funzione J, di salto di orbita, è di impedire, il più a lungo possibile, l'occorrenza di un'orbita periodica, migliorando in questo modo la caoticità delle sequenze generate.

6.1 Scelta della funzione di salto d'orbita

Si fornisce qui di seguito una preferita forma di realizzazione del metodo secondo l'invenzione, che usa la seguente funzione di salto d'orbita.

Denotando con $\mathbb{F}$ il campo ambiente (nella presente implementazione un $\mathbb{Z}_p$),

$$J:\mathbb{F}^d \to \mathbb{F}^d$$

è definita da:

$$J:\mathbb{F}^d \ni v \to \begin{pmatrix} 20\cdots 0 \\ 01\cdots 0 \\ \vdots\ \ddots\ \vdots \\ 00\cdots 1 \end{pmatrix} v + \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{F}^d$$

Ovvero esplicitamente:

$$J\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{pmatrix} := \begin{pmatrix} 2v_1 \\ v_2 + 1 \\ \vdots \\ v_d + 1 \end{pmatrix}$$

Questa preferita forma di realizzazione dell'invenzione ha il vantaggio che è molto facile da implementare perché "2" è uno shift e aggiungere "1" vuol dire cambiare la prima cifra. Oltre alla facilità di implementazione, il metodo è reso anche sensibilmente più veloce.

7. Troncamento di macchina

Il fatto che le macchine calcolatrici usuali trattano interi naturali con un numero pre-definito di bit, diciamo m (tipicamente m=32), può essere sfruttato per introdurre una non linearità supplementare che aumenta la complessità del sistema e la sua robustezza agli attacchi. Se m è il numero delle cifre binarie (precisione) disponibili per il calcolo, allora il numero primo p (modulo) è scelto in modo tale da soddisfare la seguente relazione:

$$m \geq Log(p)$$

e i coefficienti della matrice M (chiave segreta), usata per generare l'orbita, non sono presi subito

modulo p. Questo accorgimento fa sì che qualcuna delle sommatorie che intervengono nelle moltiplicazioni matrice-vettore, effettuate per la costruzione dell'orbita, possano condurre a vettori qualcuna delle cui componenti superano i $2m$ bit. Quando questo accade la macchina tronca il risultato a m bit, prima di calcolare il modulo p per M secondo lo schema:

$$\left[\left(\sum_{k=1}^{d} M[i,k]\cdot v[k]\right)\left(\mathrm{mod}\, 2^{2m}\right)\right] \ \left(\mathrm{mod}\, p\right)$$

dove $M[i,k]$ (rispettivamente $v[k]$) sono i coefficienti della matrice M (rispettivamente le componenti del vettore v).

Sebbene si sfrutti questa caratteristica delle macchine calcolatrici, il taglio può anche essere implementato come specifico calcolo; in tal caso, si sceglie un numero $q \in$ tale che:

$$q \geq Log\left(p\right)$$

e per ogni componente di vettore, nelle operazioni del metodo secondo la presente invenzione, si effettua l'operazione di modulo q prima dell'operazione modulo p, preferibilmente $q = 2^{32}$. Le due operazioni di modulo non commutano.

In tal modo l'eventuale attaccante non può sapere se, su una data componente di un dato vettore dell'orbita la legge dinamica ha agito solo con il

modulo p ovvero prima con il modulo $m=32$ e poi con il modulo p. Tenendo conto del fatto che tipicamente conviene tenere p segreto, ciò mostra che un attacco al sistema, anche solo teorico, diventa in tali condizioni non concepibile. Affinché questo troncamento sia efficace, i coefficienti delle matrici devono essere scelti abbastanza grandi (tipicamente numeri a $m-1$ o m bit) per poter generare quei resti che la macchina taglia automaticamente. L'efficacia di questa ulteriore non linearità è stata verificata sperimentalmente.

Infatti una forma realizzativa del metodo della presente invenzione, in cui tale funzione venga rimossa, non supera tutti i test statistici (seppur fornendo dei risultati apprezzabili), mentre introducendo questa operazione tutti i test vengono superati.

8. Sintesi dei passi dell'algoritmo di base

Cominciando da v_0 al primo passo 0, dopo n passi o l'algoritmo si è fermato (perché è stata calcolata la chiave crittografica per la crittazione dell'intero testo) o ha prodotto i vettori $\{v_0, v_1, ..., v_n\}$. Il passo $(n+1)$-esimo dell'algoritmo è il seguente:

(i) paragonare $k_n(v_1, ..., v_n)$ alla lunghezza del testo l_r (questo passo non si effettua nel caso di crittografia di flusso di dati);

(ii-a) si arresta se

$$k_n(v_n,\ldots,v_{n-1},\ldots,v_0)\geq l_r \qquad (2)$$

(ii-b) altrimenti calcola

$$v_{n+1} := Mv_n \quad (\text{modulo } p)$$

(iii) verifica se

$$v_{n+1} \neq v_0 \qquad (3)$$

(iv-a) se questo accade, va al passo (v);

(iv-b) se (3) non accade, definisce

$$v_{n+1} := J(v_n) \qquad (4)$$

(v) calcola la chiave relativa all'$(n+1)$-passo

$$k_{n+1}(v_1,\ldots,v_{n+1})$$

(vi) itera il procedimento ricominciando dal punto (i) con la nuova sequenza.

Nel caso di crittografia di flussi di dati, ovviamente la crittazione si effettua finché il flusso continua. Nel caso del metodo implementato su una macchina calcolatrice, il comando per iniziare la crittazione viene dato a mano oppure attivato tramite

un meccanismo di rilevamento di flusso di dati. In questa condizione, quando il flusso si interrompe, si interrompe anche la crittazione. Il computer si mette in stand-by e poi, nel caso il flusso di dati riprenda, la crittazione ricomincia utilizzando un accorgimento che verrà descritto più in là nella presente descrizione.

9. Costruzione ricorsiva della successione (k_n)

Nello schema generale dei protocolli QP-DYN le funzioni $k_n : \mathbb{N}^{dn} \to \mathbb{N}$ (KGF) possono essere arbitrarie. Esistono molte classi computazionalmente interessanti di KGF binarie d-dimensionali.

La scelta di tali funzioni può essere utilizzata:

(i) per personalizzare ulteriormente l'algoritmo;

(ii) per aumentarne la robustezza mantenendo segreta tale scelta.

Nella sezione seguente descriveremo una scelta preferita, nel senso che molte altre scelte possono essere realizzate come varianti di questa.

Tale scelta è basata sull'osservazione che un modo computazionalmente efficiente per costruire la successione (k_n) consiste nel calcolare ricorsivamente ciascuna k_n fissando una funzione:

$$k : \mathbb{N}^d \times \mathbb{N} \to \mathbb{N}$$

e definendo la KGF al primo passo $k : \mathbb{N}^d \to \mathbb{N}$ mediante la prescrizione:

$$k_1 : x \in \mathbb{N}^d \to k_1(x) := k(x,0) \in \mathbb{N}$$

La successione (k_n) viene poi definita induttivamente nel modo seguente:

$$k_{n+1} : (x,y) \in \mathbb{N}^d \times \mathbb{N} \to k_{n+1}(x,y) := k(x, k_n(y)) \in \mathbb{N}$$

Ricordando che un intero naturale è identificato ad una stringa binaria, la SSK all'(n+1)-esimo passo (cioè $k_{n+1}(v_{n+1}, v_n)$) è ottenuta combinando la SSK all'n-esimo passo (cioè $k_n(v_n, v_{n-1})$) con il nuovo dato prodotto dall'algoritmo all'(n+1)-esimo passo (cioè v_{n+1}).

Definizione 2. *Una funzione* $k : \mathbb{N}^d \times \mathbb{N} \to \mathbb{N}$ *che soddisfa la condizione:*

$$k(x,n) \geq k(0,n) \geq n \quad ; \quad \forall x \in \mathbb{N}^d \; ; \; \forall n \in \mathbb{N} \qquad (5)$$

sarà detta una KGF binaria d-dimensionale.

9.1 KGF per accodamento

Come di consueto identificheremo un intero naturale alla stringa di 0 e di 1 definita dalla sua espansione binaria. Useremo interi di b bit, con

$$b \in 32\mathbb{N}$$

cioè b è un multiplo di 32.

Definizione 3. *Data una funzione*:

$$\lambda : \{\textit{interi naturali di } b \textit{ bits}\} \to \bigcup_{m=1}^{b} \{0,1\}^m$$

che associa, ad un intero naturale di b bit, una sotto-stringa della sua espansione binaria) la funzione di accodamento sinistro:

$$k :_{\lambda} \mathbb{N}^d \times \mathbb{N} \to \mathbb{N}$$

è definita da

$$k_\lambda(n_1,\ldots,n_d;n) := \left[\lambda(n_d)\ldots\lambda(n_1),n\right] \quad ; \quad n_1,\ldots,n_d,n \in \mathbb{N} \qquad (6)$$

dove il membro destro denota la stringa binaria ottenuta accostando nel modo indicato le stringhe binarie $\lambda(n_d),\ldots,\lambda(n_1),n$.

Si osserva che la funzione di accodamento sinistro k_λ, definita dalla relazione (6), dipende dalla scelta della funzione λ. Le due scelte preferite secondo la presente invenzione sono:

(i) il troncamento casuale: si taglia il segmento di bit compreso tra i primi due 1 (inclusi) di ogni componente del vettore partendo da sinistra;

(ii) il troncamento deterministico di ordine c: si tagliano i primi c bit di ogni componente del vettore partendo da sinistra.

Il ruolo del taglio dei bit è quello di scoraggiare la ricerca esaustiva.

Tenere segreta la funzione λ aumenta la sicurezza dell'algoritmo.

10. Il protocollo a 2 primi

Il vantaggio computazionale del protocollo descritto nella la sezione 8 consiste nel fatto che la non linearità è dovuta soltanto all'operazione consistente nel ridurre $mod(p)$ il risultato di una funzione lineare. Le operazioni di taglio, salto di orbita e taglio di macchina introducono ulteriori non linearità che, da sole, rendono impossibile concepire un credibile attacco al metodo.

Un ulteriore rafforzamento della sicurezza si può ottenere replicando il sistema dinamico originale. A tal fine ci sono due principali scelte possibili:

(i) replicare la legge dinamica (matrice) lasciando l'ambiente, cioè il campo $\mathbb{Z}_p$, invariato;

(ii) replicare l'ambiente, cioè il campo $\mathbb{Z}_p$, lasciando la legge dinamica invariata.

Dal punto di vista qualitativo entrambi i casi conducono a sistemi dinamici differenti, cioè a orbite differenti. Poiché la scelta (ii) presenta il vantaggio, rispetto al calcolo e alla memoria, di utilizzare ad ogni punto dell'iterazione una sola matrice, scelta tale scelta risulta preferita secondo l'invenzione. In quanto segue illustreremo questa

realizzazione forma realizzativa del metodo secondo l'invenzione.

Inoltre, poiché il livello di sicurezza realizzato con una semplice duplicazione è abbastanza accettabile, limiteremo la nostra discussione a questo caso, sebbene sia possibile replicare il sistema dinamico quante volte si vuole seguendo la stessa procedura.

Dato un testo T di lunghezza pari a l_T bit, consideriamo due sistemi dinamici:

$$\{d, M, p', v_0, k_n, J\} \;\; ; \;\; \{d, M, p'', v_0, k_n, J\} \qquad (7)$$

con:

(i1) la stessa dimensione $d \in \mathbb{N}$

(i2) la stessa legge dinamica $M \in M(\mathrm{d};\, \mathbb{N})$

(i3) lo stesso vettore iniziale $v_0 \in \mathbb{N}^d$

(i4) la stessa funzione di salto di orbita $J : \mathbb{N}^d \times \mathbb{N}^d \to \mathbb{N}^d$

(i5) la stessa funzione generatrice della chiave (KGF) $k_n : \mathbb{N}^d \to \mathbb{N}$ $(n \in \mathbb{N})$

(i6) due numeri primi $p', p \in \mathbb{N}$ detti "moduli".

Poi si esegue il metodo descritto nella sezione 8 fino al punto (v) escluso (calcolo della $(n+1)$-ma chiave), che viene modificato come segue.

Avendo prodotto due chiavi all'$(n+1)$-mo passo:

$$k_{n+1}(v_1, \ldots, v_{n+1}) \;\; ; \;\; k_{n+1}(v'_1, \ldots, v'_{n+1})$$

(si ricordi che le KGF k_n sono le stesse) l'algoritmo calcola:

$$k_{n+1}(v_1,\ldots,v_{n+1})\oplus k_{n+1}(v'_1,\ldots,v'_{n+1}) \qquad (8)$$

dove, per due qualsiasi stringhe binarie x, y, $x\oplus y$ denota la stringa $x\mathrm{XOR}y$ e se necessario le due stringhe sono rese della stessa lunghezza aggiungendo degli zeri a sinistra.

Poi il metodo rimuove dai bit di (8) tutti gli 0 principali (corrispondenti cioè alle potenze più alte di 2) ed il primo 1.

Il risultato è la chiave all'$(n+1)$-esimo passo dell'algoritmo modificato:

$$\overline{k_{n+1}}(v_1,\ldots,v_{n+1};v_1',\ldots,v_{n+1}')$$

La regola d'arresto delle fasi del metodo è la stessa di quella della sezione 8.

Si osserva che può accadere che la stringa (8) abbia alcuni dei bit principali (potenze superiori di 2) uguali a zero, perché le operazioni di modulo p e p' entrano in gioco solo dopo un certo numero di punti dell'orbita, e quindi, in questi passi, le due chiavi parziali sono identiche.

Questa considerazione riguarda la procedura di inizializzazione, in cui si definisce v_0. v_0 è pubblico, mentre la matrice M è segreta ed è essa che fa perdere

memoria dello stato iniziale, in modo da proteggere il testo crittato contro un attaccante.

Si può certamente partire non da v_0, ma da una funzione arbitraria $f(v_0)$, che è tenuta segreta.

In aggiunta, o in alternativa, si possono scartare i primi h passi dell'orbita così che, non conoscendo la legge del moto, si perde memoria di v_0.

La procedura di inizializzazione è in ogni caso arbitraria.

11. Attacco al caso di una matrice e considerazioni di complessità

Si fornisce qui una prova di robustezza del metodo secondo la presente invenzione. Le analisi di robustezza che seguono sono state sviluppate nelle ipotesi peggiori possibili per il difensore, ovvero:

- si considera il caso a una sola matrice (escludendo quindi il più rilevante fattore di sicurezza);
- si esclude il taglio di macchina;
- si suppone che l'unica chiave segreta sia la matrice M mentre sono considerate pubbliche le informazioni seguenti:
- il numero primo p (modulo),
- la dimensione d,
- il dato iniziale v_0,
- le KGF k_n,
- il taglio dei bit (si veda la fine della sezione 9.1) è considerato fisso e pubblico.

Oltre a ciò si considera il caso più favorevole per l'attaccante, ovvero *l'attacco a testo in chiaro*, in cui sia il testo originario T, che il testo codificato sono noti all'attaccante, e quindi la SSK è nota (la quale viene però cambiata ad ogni messaggio!). Ci si chiede allora: l'attaccante può ricostruire la chiave segreta ovvero la matrice M?.

L'attaccante, nel seguito, sarà denotato con E. Il grado di inviolabilità naturalmente cresce enormemente se, come è sempre possibile, qualcuna di queste informazioni è tenuta segreta (e condivisa). Infatti, conoscendo T e $C = T\ xor\ S$ e usando l'idempotenza dell'operazione xor, E può recuperare $S = C\ xor\ T$.

Ciò aiuterà a intuire perché finora la Richiedente non sia riuscita a trovare, neppure a livello teorico, attacchi alla versione a 2 matrici dell'algoritmo.

Si suppone:

(i) che E conosca $d+1$ vettori consecutivi dell'orbita;

(ii) che i primi d tra tali vettori siano linearmente indipendenti e si definiscono le seguenti matrici per accostamento di vettori colonna:

$$V = \left(v_l \middle| \cdots \middle| v_{l+d-1}\right) \in M\ (\mathrm{d};\ \mathbb{N}) \quad : \quad V' = \left(v_{l+1} \middle| \cdots \middle| v_{l+d}\right) \in M\ (\mathrm{d};\ \mathbb{N})$$

allora $MV = V'$ e ciò permette di ottenere $M = V'\ V^{-1}$.

Tuttavia E ha l'intera stringa S, ma non la corrispondente suddivisione indicata dall'equazione (9) (si veda più oltre), necessaria per recuperare le componenti dei vettori v_i. E deve tentare di indovinare

quale sia la suddivisione corretta. A tal fine E può sfruttare la sua conoscenza della struttura dell'algoritmo, che è pubblica, vale a dire la costruzione:

- calcolo iterativo dei vettori dell'orbita: $v_0, v_1, v_2, \ldots$ usando M come descritto sopra;
- per ogni componente $j=1,\ldots,d$ di ciascun vettore v_i, considerare la corrispondente rappresentazione in base 2 con c cifre binarie:

$$v_{ij} = \overbrace{0\ldots01}^{d_{ij}}\ \overbrace{X\ldots X}^{u_{ij}} \qquad (c\ digits)$$

nella quale i principali 0 e 1 sono messi in evidenza, e i rimanenti bit

$$u_{ij} = Log(v_{ij}) - 1$$

sono genericamente rappresentati con la lettera $X \in \{0,1\}$;

- genera la sequenza pseudocasuale giustapponendo iteratamente i bit principali di d_{ij} e giustapporre dopo avere effettuato il taglio (fisso o casuale):

$$S = \underbrace{X\ldots X}_{u_{i1}} \Big| \underbrace{X\ldots X}_{u_{i2}} \Big| \ldots \Big| \underbrace{X\ldots X}_{u_{id}} \Big| \underbrace{X\ldots X}_{u_{i+k,1}} \Big| \underbrace{X\ldots X}_{u_{i+k,d}} \qquad (9)$$

- non appena sono ottenuti n bit, il processo è interrotto e la sequenza pseudocasuale è restituita.

Dato che il taglio dei bit è considerato fisso, diciamo pari a τ bit, e pubblico, E sa che, per ogni componente di ogni vettore, perde $d_{ij} = \tau$ bit e quindi ha una ambiguità di 2^{τ} possibilità per ogni componente. Dato che tali componenti sono d, l'ambiguità è di $2^{d\tau}$ possibilità per ogni vettore. Poiché E ha bisogno di $d+1$ vettori, egli si trova a dover scegliere tra $2^{d(d+1)\tau}$ possibilità. Per esempio, se $d=10$ (dimensione che un comune personal può gestire senza alcuna difficoltà), allora $d(d+1)=110$. Supponendo, per facilitare ulteriormente il lavoro di E, che $\tau=2$, E si trova a dover scegliere tra 2^{220} possibilità. Per ciascuna di queste scelte E deve effettuare una inversione e una moltiplicazione di matrici di ordine 10. Si osservi infine che un incremento di d o di τ, per esempio rispettivamente da 10 a 15 e da 2 a 3, aumenta la difficoltà di costruzione per un fattore che è al più quadratico nell'incremento, mentre la difficoltà di attacco aumenta esponenzialmente.

11.1 Attacco al caso di 2 primi

Se l'algoritmo crittografico usato è quello a 2 primi, le considerazioni descritte nella sezione precedente sono completamente distrutte, anche in assenza di tagli, dal fatto che, con questo algoritmo, E può recuperare soltanto la sequenza:

$$\overline{k_N}\,(v_1,...,v_N;v'_1,...v'_N) := k_N(v_1,...,v_N) \oplus k_N(v'_1,....,v'_N)$$

(dove $\oplus$ = *xor*) ma è impossibile sapere se, in tale sequenza, un 1 è stato ottenuto dalla combinazione di uno 0 in $k_N(v_1,...,v_N)$ (SSK del primo sistema dinamico) e di un 1 in $k_N(v'_1,...,v'_N)$ (SSK del secondo sistema dinamico), o viceversa. Similmente è impossibile sapere se uno 0 sia derivato da due 0 o da due 1.

In altre parole, e questa è una delle idee principali del metodo secondo l'invenzione, *E* non si trova di fronte a un problema difficile, bensì di fronte a un problema indeterminato del tipo: data una somma di due numeri ricostruire il valore esatto degli addendi. Poiché, fissando arbitrariamente uno dei due numeri, la conoscenza della somma determina univocamente l'altro e poiché, rispetto alle informazioni di E tutti i numeri del campo di base sono equiprobabili, ne segue che, per ogni componente del vettore, E ha una ambiguità dello stesso ordine del numero di elementi del campo, cioè p. Per un vettore l'ambiguità sarà quindi dell'ordine di p^d e per d+1 vettori dell'ordine di $p^{d(d+1)}$. Infine, l'uso simultaneo dei tre campi differenti $\mathbb{Z}_{p'}$, $\mathbb{Z}_{p''}$, $\mathbb{Z}_2$ (dove l'ultimo si riferisce all'operazione di *xor*), rende praticamente impossibile una estrazione di informazioni, anche solo statistiche, mediante metodi algebrici.

Bibliografia

[AcdeTiDiLi81]
Accardi L., F. de Tisi, A. Di Libero:
Sistemi dinamici instabili e generazione di successioni pseudo-casuali,In: Rassegna di metodi statistici e applicazioni, W. Racugno (ed.) Pitagora Editrice, Bologna (1981) 1-32

[AbAcAu91]
Abundo M., Accardi L., Auricchio A.:
Hyperbolic automorphisms of tori and pseudo-random sequences, Calcolo 29 (1992) 213-240

[AcRe93]
Accardi L., Regoli M.:
Some simple algorithms for forms generations, IN: L. Accardi (ed.), Fractals in nature and in mathematics, Acta Encyclopaedica, Istituto dell'Enciclopedia Italiana (1993) 109-116

[Cugiani80]
M.Cugiani: Metodi Numerico Statistici (1980)

[Gäbler07]
Markus Gäbler:
Statistical Analysis of Random Number Generators
Rapporto interno del Centro V. Volterra, Ottobre (2007)

[ItaOttGrilLent09]

Giuseppe F. Italiano, Vittorio Ottaviani, Antonio Grillo, Alessandro Lentini: BENCHMARKING FOR THE QP CRYPTOGRAPHIC SUITE, Agosto (2009)

[REG10]
Regoli, M., A redundant cryptographic symmetric algorithm that confounds statistical tests,
Proceeding International Conference QBIC10, World Scientific, Series QP--PQ (2010)

[RobshBil08]
Mattew Robshaw, Olivier Billet (Eds.): New Stream Cipher Designs, The eSTREAM Finalists, State-of-the-Art Survey, LNCS 4986 Springer (2008).

In quel che precede sono state descritte le preferite forme di realizzazione e sono state suggerite delle varianti della presente invenzione, ma è da intendersi che gli esperti del ramo potranno apportare modificazioni e cambiamenti senza con ciò uscire dal relativo ambito di protezione, come definito dalle rivendicazioni allegate.

Barzanò & Zanardo Roma S.p.A.

RIVENDICAZIONI

1. Metodo di crittografia per lo scambio di dati tra due interlocutori, che utilizza i vettori di un sistema dinamico a tempo discreto determinato da:

(i1) un intero naturale $d \in \mathbb{N}$, ove $\mathbb{N}$ è l'insieme dei numeri naturali positivi, chiamato "dimensione dell'algoritmo";

(i2) una matrice quadrata $\mathbf{M}$ a coefficienti interi naturali e di dimensione $d \times d$, che rappresenta la legge del moto del sistema dinamico;

(i3) un intero naturale $p \in \mathbb{N}$, chiamato "modulo", preferibilmente un numero primo;

(i3) un intero naturale $n \in \mathbb{N}$, che rappresenta il contatore del passo n-esimo di iterazione della legge del moto;

(i4) un vettore iniziale v_0, ed un vettore $v_n = M^n v_0$ relativo al passo di iterazione n-esimo,

per generare una chiave binaria di crittazione k, definendo, per ogni $n \in \mathbb{N}$, una funzione $k_n : \mathbb{N}^{dn} \to \mathbb{N}$, denominata "funzione generatrice della chiave di ordine n" o "KGF" che usa l'espansione binaria delle componenti degli n vettori prodotti nei primi n passi di interazione della legge del moto per costruire la chiave parziale n-esima, il metodo comprendendo l'esecuzione delle seguenti fasi successive:

A. cominciando da v_0 al primo passo 0, calcolare n passi di iterazione della legge del moto calcolando sempre le componenti dei vettori modulo

p, avendo così all'n-esimo i vettori $\{v_0, v_1, \ldots, v_n\}$;

B. calcolare il vettore al passo $(n+1)$-esimo:

$$v_{n+1} := Mv_n \quad (\text{modulo } p)$$

C. calcolare la chiave parziale $(n+1)$-esima:

$$k_{n+1}(v_1, \ldots, v_{n+1})$$

D. procedere al calcolo della funzione KGF al passo successivo ripetendo le fasi B e C, fino al passo $n = H \in \mathbb{N}$ in cui la chiave crittografica $k_H = k$ ha la lunghezza desiderata;

il metodo essendo caratterizzato dal fatto che:

I. nella fase C, data una funzione:

$$\lambda : \{\textit{interi naturali di b bits}\} \to \bigcup_{m=1}^{b} \{0,1\}^m$$

con $b \in \mathbb{N}$ e che associa ad un intero naturale di b bit una sotto-stringa della sua espansione binaria, la KGF è la funzione $k_\lambda : \mathbb{N}^d \times \mathbb{N} \to \mathbb{N}$ e $k_{n+1} = k_\lambda(n_1, \ldots, n_d; k_n)$, $n_1, \ldots, n_d \in \mathbb{N}$, $n_1, \ldots, n_d$ essendo le componenti del vettore v_{n+1};

II. si sceglie un numero $q \in \mathbb{N}$ tale che:

$$q \geq Log(p)$$

e per ogni componente di vettore, nelle fasi A e B, si effettua un'operazione di modulo q prima della suddetta operazione modulo p, preferibilmente con $q = 2^{32}$;

III. nella fase B:

- si confronta il vettore corrente v_{n+1} con il vettore iniziale v_0;
- se $v_{n+1} \neq v_0$, si passa direttamente alla fase C;
- se $v_{n+1} = v_0$, si ridefinisce:

$$v_{n+1} := J(v_n)$$

dove $J:\mathbb{N}^d \to \mathbb{N}^d$ è una funzione di salto d'orbita, e si passa alla fase C, l'operazione essendo equivalente a ricominciare una nuova orbita dal vettore $J(v_n)$, che quindi diventa il nuovo punto iniziale;

IV. si effettuano:

- le fasi da A a C per almeno due sistemi dinamici definiti da corrispondenti differenti insiemi $\{d, M, p, k_n, J\}$ e $\{d', M', p', k'_n, J'\}$ che danno luogo ad almeno due rispettive chiavi parziali $k_{n+1}(v_1,\ldots,v_{n+1})$ e $k'_{n+1}(v'_1,\ldots,v'_{n+1})$; e
- alla fine della fase C la chiave parziale $(n+1)$-esima è la chiave parziale combinata $\overline{k_{n+1}}(v_1,\ldots,v_{n+1};v_1',\ldots,v_{n+1}')$ ottenuta effettuando

l'operazione di XOR tra dette almeno due chiavi parziali, dove, se dette almeno due chiavi parziali sono di lunghezza differente, si rendono della stessa lunghezza aggiungendo o eliminando cifre, ed n può essere arbitrario oppure $n = H$.

2. Metodo secondo la rivendicazione 1, caratterizzato dal fatto che le matrici M e M' di detti almeno due sistemi dinamici sono invertibili e selezionano pertanto corrispondenti sistemi dinamici reversibili.

3. Metodo secondo la rivendicazione 1 o 2, caratterizzato dal fatto che la KGF è la funzione di accodamento sinistro:

$$k_\lambda : \mathbb{N}^d \times \mathbb{N} \to \mathbb{N}$$

definita da:

$$k_{n+1} = k_\lambda(n_1,\ldots,n_d;k_n) := \left[\lambda(n_d),\ldots\lambda(n_1),s\right] \quad ; \quad n_1,\ldots,n_d,s \in \mathbb{N}$$

dove il membro più a destra denota la stringa binaria ottenuta accostando nel modo indicato le stringhe binarie $\lambda(n_d),\ldots,\lambda(n_1),n$, $n_1,\ldots,n_d$ essendo le componenti del vettore v_{n+1}.

4. Metodo secondo una qualsiasi delle rivendicazioni da 1 a 3, caratterizzato dal fatto che nell'operazione I detta sotto-stringa risulta da un taglio di un numero di cifre, partendo dai bit di potenza maggiore, pari ad un numero intero $\tau_{n,v_{nj}}$

variabile da 0 a q e che dipende da n e dalla j-esima componente del vettore v_n, ottenendo così una chiave k con $\sum_{n=1}^{N}\sum_{j=1}^{d}(q-\tau_{n,v_{nj}})$ bit.

5. Metodo secondo la rivendicazione 4, caratterizzato dal fatto che nell'operazione I detta sotto-stringa risulta da un troncamento deterministico di ordine τ, τ essendo un numero intero positivo, che consiste nel taglio dei primi $\tau_{n,v_{nj}} = \tau$ bit di ogni componente del vettore v_n partendo da sinistra o da destra.

6. Metodo secondo la rivendicazione 4, caratterizzato dal fatto che, nell'operazione I detta sotto-stringa risulta dal taglio dei primi zeri da sinistra fino al primo "1" incluso di ogni componente del vettore v_{n+1} partendo da sinistra o da destra, e/o, alla fine dell'operazione IV, la chiave parziale combinata risulta da detta operazione di XOR e dal successivo taglio dei primi zeri da sinistra fino al primo "1" incluso partendo da sinistra o da destra.

7. Metodo secondo una qualsiasi delle rivendicazioni da 1 a 6, caratterizzato dal fatto che la funzione $J:\mathbb{N}^d \to \mathbb{N}^d$ è definita da:

$$J:\mathrm{F}^d \ni v \to \begin{pmatrix} 20\cdots0 \\ 01\cdots0 \\ \vdots\ \vdots\ \ddots\ \vdots \\ 00\cdots1 \end{pmatrix} v + \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{N}^d$$

ovvero, esplicitamente:

$$J\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{pmatrix} := \begin{pmatrix} 2v_1 \\ v_2 + 1 \\ \vdots \\ v_d + 1 \end{pmatrix},$$

dove l'uguaglianza è intesa modulo p.

8. Metodo secondo una qualsiasi delle rivendicazioni da 1 a 7, caratterizzato dal fatto che nell'operazione IV il campo ambiente è duplicato, ovvero $d = d'$, $M = M'$ e $p \neq p'$, ed inoltre $J = J$ e $k_n = k'_n$.

9. Metodo secondo una qualsiasi delle rivendicazioni da 1 a 8, in cui la chiave privata ha una parte potenzialmente pubblica ed una segreta, caratterizzato dal fatto che la parte potenzialmente pubblica è costituita dal vettore iniziale v_0, e la parte segreta, anche detta parte simmetrica, è costituita dalla matrice M, mentre la SSK è la chiave crittografica k, detti due interlocutori cambiando arbitrariamente la parte potenzialmente pubblica della chiave privata al fine di ottenere, per differenti comunicazioni di dati, di volta in volta delle chiavi segrete condivise SSK differenti.

10. Metodo secondo una qualsiasi delle rivendicazioni da 1 a 9, caratterizzato dal fatto di comprendere una procedura di inizializzazione, in cui il vettore iniziale v_0 è arbitrariamente definito, attraverso una funzione arbitraria $f(v_0)$, che è tenuta segreta, e/o si scartano i primi $h \in \mathbb{N}$ passi dell'orbita.

11. Programma per elaboratore caratterizzato dal fatto di comprendere mezzi a codice atti ad eseguire, quando operano su un elaboratore, il metodo secondo una qualsiasi delle rivendicazioni da 1 a 10.

12. Supporto di memoria leggibile da un elaboratore, avente un programma memorizzato su di esso, caratterizzato dal fatto che il programma è il programma per elaboratore secondo la rivendicazione 11.

Barzanò & Zanardo Roma S.p.A.