Appendice

Alcuni algoritmi crittografici

Elenchiamo alcuni fra gli algoritmi crittografici di maggior interesse utilizzati nei moderni sistemi informatici di protezione. Per ulteriori informazioni sulle specifiche progettuali, sui dati tecnici e sui sorgenti di alcuni di essi, è possibile fare riferimento ai documenti contenuti nel CD-ROM allegato al libro.

Cifrari a blocchi

3-Way

È un semplice e veloce cifrario a blocchi (block cipher) sviluppato da Joan Daemen. Utilizza una chiave di 96 bit su blocchi di dati della stessa lunghezza. Usa una procedura iterativa con alcune semplici operazioni da applicare ai dati in chiaro per un numero specificato di volte. David Wagner, John Kelsey e Bruce Schneier della Counterpane Systems hanno scoperto un attacco sulle chiavi di questo sistema mediante interrogazioni su circa 222 testi in chiaro/cifrati. L'algoritmo 3-Way non è registrato.

Blowfish

Blowfish è un cifrario a blocchi sviluppato da Bruce Schneier, autore del famoso libro *Applied Cryptography*. Questo algoritmo utilizza varie tecniche tra le quali la rete Feistel, le S-box dipendenti da chiavi e funzioni F non invertibili che lo rendono, forse, l'algoritmo

più sicuro attualmente disponibile. Non si conoscono al momento attacchi nei suoi confronti.

CAST

L'algoritmo CAST, progettato da Carlisle Adams e Stafford Taveres, è ottimo e molto stabile. Molto simile al Blowfish come struttura, poiché utilizza più o meno le stesse tecniche crittografiche (con l'eccezione della rete Feistel rimpiazzata da un sistema chiamato di "permutazioni-sostituzioni"). David Wagner, John Kelsey e Bruce Schneier hanno scoperto un attacco sulle chiavi a 64 bit del CAST, mediante 217 testi cifrati, con 248 computazioni dell'algoritmo. Naturalmente l'attacco non è efficace al 100 per cento. L'algoritmo CAST è registrato dalla Entrust Technologies, che lo ha rilasciato per un uso libero e gratuito.

CMEA

È un algoritmo di cifratura sviluppato dalla Telecommunications Industry Association per l'utilizzo nella telefonia cellulare. Utilizza chiavi a 64 bit con caratteristiche di variabilità della lunghezza del blocco. Questo algoritmo viene usato per la cifratura dei canali di controllo dei cellulari. Si distingue dall'ORYX, un sistema altrettanto insicuro di cifratura a blocchi, utilizzato anch'esso nelle comunicazioni cellulari. Il CMEA è stato violato da
David Wagner, John Kelsey e Bruce Schneier della Counterpane Systems.

DES

Sviluppato dall'IBM nel 1970 e successivamente ufficializzato come algoritmo standard di cifratura dei documenti non classificati, nel 1976 il DES è diventato uno dei punti di riferimento per il commercio crittografico. È riuscito a rimanere efficace fino a pochi anni fa. Il DES si basa sull'utilizzo di chiavi a 64 bit, dei quali solo 56 effettivamente utilizzati per la generazione della chiave (gli altri 8 servono per le operazioni di correzione degli errori). Ancora in uso attualmente in molti sistemi crittografici ritenuti ovviamente insicuri. Durante lo sviluppo del DES, l'NSA inserì sistemi segreti S-box nel

codice, che, dopo diverse crittanalisi, si sono rilevati robusti. Attualmente infatti non esistono ancora studi sulla violazione strutturale di questo algoritmo, che è stato rotto soltanto grazie all'enorme crescita della potenza di calcolo dei moderni elaboratori. Recentemente, il 17 luglio 1998, l'EFF (Electronic Frontier Foundation) è riuscita a implementare una scheda multiprocessore in grado di violare un sistema DES a 64 bit in meno di tre giorni, generando tutte le 2^56 chiavi possibili (figura A.1). Questa scheda è basata sull'utilizzo di alcuni chip progettati sempre dall'EFF, chiamati "Deep Crack", in grado di generare 88 bilioni di chiavi DES al secondo (figura A.2 e A.3). Si possono trovare le specifiche di realizzazione del progetto "Deep Crack" e della scheda multiprocessore di cracking al seguente indirizzo web: `http://www.eff.org` oppure nel libro dell'EFF *Cracking DES, Secrets of Encryption Research, Wiretap Politics & Chip Design*.

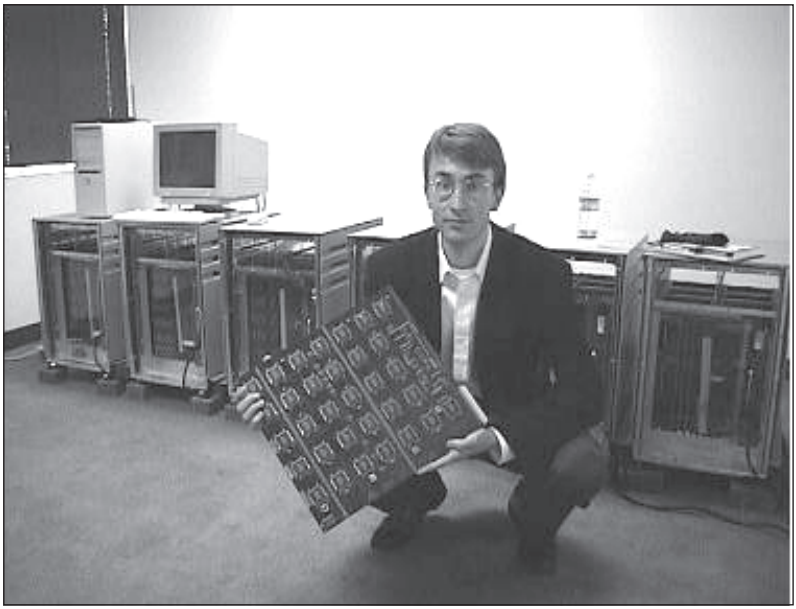

Figura A.1

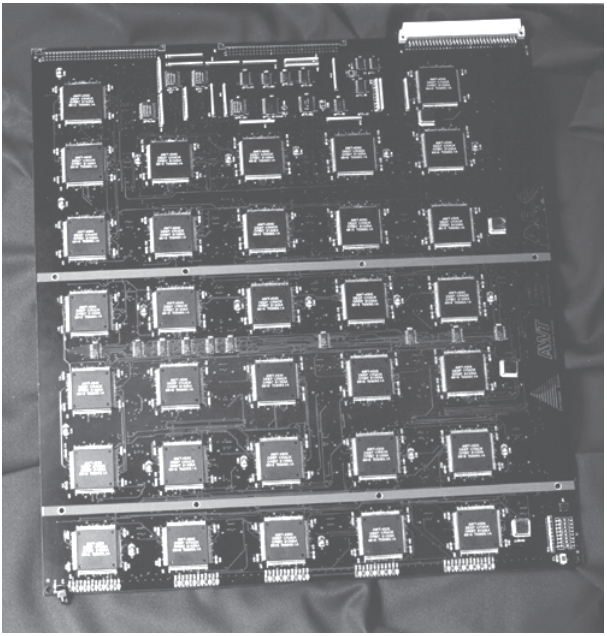

Figura A.2

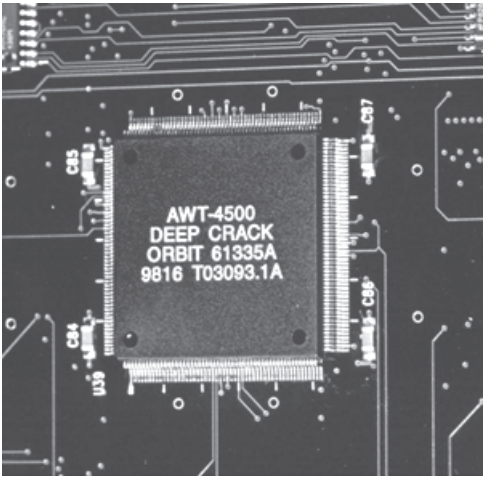

Figura A.3

FEAL
Sviluppato dalla Nippon Telephone & Telegraph come rimpiazzo all'algoritmo DES, il Fast Data Encipherment Algorithm (FEAL) è molto insicuro. FEAL-4, FEAL-8 e FEAL-N sono stati tutti soggetti a varie crittanalisi che sono riuscite a violare il sistema con soli 12 testi di riferimento cifrati. L'algoritmo FEAL è registrato.

GOST
GOST è un algoritmo crittografico sviluppato in Russia e utilizzato in tutta l'unione sovietica come clone del DES, dal punto di vista sia tecnico sia politico. L'algoritmo utilizza 32 cicli iterativi per l'opera-zione di cifratura con chiavi di 256 bit. La forza del GOST sembra essere legata proprio all'estrema segretezza dei suoi dettagli implementativi. John Kelsey nel febbraio 1996 ha reso noto un attacco a questo algoritmo, pubblicando il materiale nella mailing-list `sci.crypt`.

IDEA
È stato sviluppato a Zurigo in Svizzera da Xuejia Lai e James Massey. Ritenuto il migliore e il più affidabile algoritmo a blocchi tradizionale, è disponibile nel circuito del pubblico dominio. Utilizza chiavi di 128 bit e ha resistito fino a questo momento a qualsiasi attacco di crittanalisi. IDEA è registrato dalla società Ascom.

LOKI
Sviluppato come possibile rimpiazzo del DES, l'algoritmo utilizza chiavi di 64 bit su blocchi della stessa lunghezza. La prima versione di questo algoritmo è stata analizzata con studi di crittanalisi che hanno rilevato una complementarietà di 8 bit (questo significa che il numero di chiavi da utilizzare in un attacco a forza bruta è ridotto a 256). LOKI è stato riprogettato e rilasciato successivamente con il nome LOKI91. Test di crittanalisi non ne hanno dimostrato l'insicurezza, anche se sono allo studio attacchi di crittanalisi differenziale che potrebbero dare altri risultati.

Lucifer
È stato uno dei primi algoritmi moderni di crittografia. Sviluppato dall'IBM nel 1960 dal ricercatore Horst Feistel, Lucifer è oggi considerato il precursore del DES. Esistono diverse manipolazioni e "reincarnazioni" di questo algoritmo con nomi simili e tutte sono insicure, compreso l'originale. Un libro sulla crittanalisi differenziale di Lucifer è stato scritto dai ricercatori Ishai Ben-Aroya e Eli Biham.

MacGuffin
Si tratta di un cifrario sviluppato da Matt Blaze e Bruce Schneier, come esperimento di design. Il MacGuffin utilizza una rete di Feistel, senza l'operazione di "split", dividendo i blocchi di 64 bit in due parti, rispettivamente di 16 bit e 48 bit (questo metodo è chiamato Generalized Unbalanced Feistel Network, GUFN). Un attacco di crittanalisi differenziale è stato trovato utilizzando 251,5 testi cifrati.

MARS
Sviluppato dall'IBM, MARS utilizza blocchi di 128 bit per le operazioni di cifratura e supporta chiavi di lunghezza variabile da 128 a 1248 bit. Questo algoritmo è davvero unico, dal momento che utilizza praticamente tutti i sistemi crittografici conosciuti per l'operazione di cifratura/decifratura. Addizioni, sottrazioni, S-box, rotazioni a virgola fissa e a virgola mobile, prodotti eccetera.

MISTY
È un algoritmo crittograffico sviluppato dalla Mitsubishi Electric dopo la violazione del DES, nel 1994. Concepito per resistere a crittanalisi lineari e differenziali, risulta ancora in fase di test anche perché molto segreto. È stato preso in considerazione come aggiunta allo standard SET 2.0.

MMB
È stato progettato come alternativa all'IDEA; infatti utilizza gli stessi concetti progettuali. Sfortunatamente non è altrettanto sicuro e sono

stati portati a termine con successo numerosi attacchi nei suoi confronti.

NewDES
Sviluppato come alternativa al DES da Robert Scott, il NewDES ha avuto una durata molto breve. È stato violato con l'utilizzo di 24 chiavi probabili e 530 testi cifrati.

RC2
Come l'RC4, l'RC2 è nato come cifrario segreto. Successivamente è apparso nei dettagli implementativi nella mailing list sci.crypt.
David Wagner, John Kelsey, e Bruce Schneier hanno scoperto un attacco dell'RC2 basato su chiavi, con la conoscenza approssimativa di 234 testi cifrati. L'RC2 non è registrato dall'RSA Data Security, è solo protetto da segreto.

RC5
Si tratta di un gruppo di algoritmi, sviluppati da Ron Rivest dell'RSA Data Security, che lavorano su blocchi di dati, chiavi e numeri casuali di lunghezza variabile. La lunghezza del blocco dipende generalmente dalla lunghezza della word utilizzata dal computer sul quale l'algoritmo è stato implementato. Sui processori a 32 bit, quindi con chiavi di 32 bit, l'RC5 lavora su blocchi di 64 bit. David Wagner, John Kelsey e Bruce Schneier hanno scoperto chiavi deboli nell'RC5, che hanno una probabilità di scelta tra 2 e 10r, dove r è il numero di tentativi. Per valori r sufficientemente grandi invece (maggiori di 10), non ci sono problemi di sicurezza. Kundsen ha trovato un possibile attacco differenziale a questo algoritmo, le cui specifiche si possono trovare nei documenti dell'RSA. L'RC5 è ovviamente registrato dall'RSA Data Security, Inc.

RC6
L'RC6 è stato sviluppato sotto richiesta del Ronald Rivest's AES. Come il cifrario AES, lavora su blocchi di 128 bit e accetta chiavi di lunghezza variabile. Molto simile all'RC5, incorpora vari studi di

analisi dell'algoritmo precedente, che hanno dimostrato che non tutti i bit dei dati sono usati per determinare l'ammontare delle rotazioni; l'RC6 utilizza prodotti per il calcolo delle rotazioni e tutti i dati di input per il calcolo dell'ammontare delle stesse.

REDOC

Ci sono due versioni dell'algoritmo REDOC, chiamate rispettivamente REDOC II e REDOC III. REDOC II è considerato sicuro; è interessante poiché utilizza maschere di dati per la selezione dei valori nelle S-box. Utilizza inoltre chiavi di 160 bit e lavora su blocchi di 80 bit. REDOC III è più lento e meno sicuro, anche se utilizza chiavi lunghe fino a 20.480 bit.

Rijndael

È un algoritmo sviluppato da Joan Daemen e Vincent Rijmen sotto richiesta dell'AES. Il cifrario utilizza chiavi di lunghezza variabile (gli autori hanno dimostrato come è possibile variare le dimensioni delle chiavi con multipli di 32 bit). Lo schema del Rijndael è stato influenzato dall'algoritmo SQUARE.

Safer

È stato sviluppato da Robert Massey su richiesta della Cylink Corporation. Ne esistono molte versioni differenti, con lunghezza delle chiavi di 40, 64 e 128 bit.

Alcuni attacchi hanno dimostrato la sicurezza di questo algoritmo con la crittanalisi differenziale e lineare, anche se Bruce Schneier, autore del libro *Applied Cryptography*, raccomanda di non utilizzarlo per ragioni "politiche", poiché "Safer è stato sviluppato dalla Cylink, e la Cylink opera in stretto contatto con l'NSA."

Serpent

L'algoritmo Serpent è stato sviluppato da Ross Anderson, Eli Biham, e Lars Knudsen dietro richiesta dell'AES. Gli autori hanno utilizzato tecniche combinate per la sua creazione, ispirandosi all'algoritmo DES. Serpent utilizza blocchi di 128 bit con chiavi di 256 bit. Come il DES, contiene delle permutazioni iniziali e finali senza nessuna

giustificazione crittografica, che, apparentemente, servono per ottimizzare i dati prima dell'operazione di cifratura. L'algoritmo è stato rilasciato durante la convention "5th International Workshop on Fast Software Encryption": questa iterazione del Serpent è stata chiamata Serpent 0 e utilizza le S-box originali del DES. Dopo i commenti e le discussioni raccolti durante la convention, le permutazioni sono state cambiate e chiamate Serpent 1. Il nuovo algoritmo ha resistito alla crittanalisi differenziale e lineare.

SQUARE

È un cifrario a blocchi di tipo iterativo su insiemi di 128 bit con l'utilizzo di chiavi sempre di 128 bit. La funzione di arrotondamento di questo algoritmo è composta da quattro trasformazioni: una trasformazione lineare, una trasformazione non-lineare, una permutazione a livello di byte e una addizione sui bit con la chiave. SQUARE è stato progettato per resistere agli attacchi della crittanalisi differenziale e lineare.

Skipjack

Dopo l'insuccesso del progetto Clipper, l'NSA ha rilasciato questo algoritmo chiamato Skipjack, che è formalmente un algoritmo segreto di crittografia e utilizza chiavi di 80 bit. Esistono copie "abusive" delle sue specifiche tecniche, rintracciabili presso il sito web del NIST. Eli Biham e Adi Shamir hanno pubblicato i primi risultati di crittanalisi su questo algoritmo, ma è ancora troppo presto per affermare che è stato violato completamente.

Tiny Encryption Algorithm (TEA)

TEA è un cifrario sviluppato per ottimizzare le prestazioni in termini di spazio di memoria occupato e velocità. Queste ottimizzazioni hanno però provocato una scarsa sicurezza dell'algoritmo, a tal punto che si è riusciti a violarlo con l'utilizzo della crittanalisi differenziale con solo 223 testi cifrati. Il problema nasce dalla semplice procedura di scheduling delle chiavi. Ogni chiave TEA può essere trovata in similitudine con altre tre chiavi, come descritto nel libro di David Wagner, John Kelsey e Bruce Schneier. Questo preclude la

possibilità di utilizzare TEA come funzione hash. Roger Needham e David Wheeler hanno proposto una versione di questo algoritmo con il conteggio delle chiavi in eccesso.

Twofish
Twofish è stato sviluppato sotto richiesta della Counterpane Systems' AES. Sviluppato dal team Counterpane (composto da Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall e Niels Ferguson), questo algoritmo è stato dichiarato sicuro dopo attente analisi.

Stream Cipher

ORYX
ORYX è un algoritmo utilizzato per cifrare i dati delle comunicazioni cellulari. È un cifrario basato su tre funzioni di Galois LFSR a 32 bit. Si distingue dall'algoritmo CMEA, un cifrario a blocchi utilizzato per la protezione dei dati del canale di controllo delle comunicazioni cellulari. Il team crittografico della Counterpane Systems (composto da David Wagner, John Kelsey e Bruce Schneier) è riuscito a sviluppare un attacco all'ORYX basato sulla conoscenza di circa 24 byte di un testo cifrato con circa 216 parametri iniziali.

RC4
È un algoritmo della RSA Data Security, Inc. Originariamente le specifiche progettuali dell'RC4 erano segrete, poi sono state divulgate in modo anonimo nel 1994. Questo algoritmo è molto utilizzato in vari tipi di applicazioni. Non ci sono attacchi conosciuti nei suoi confronti. La versione internazionale dell'RC4 a 40 bit è stata violata con il metodo a forza bruta.

SEAL
Sviluppato da Don Coppersmith dell'IBM Corp, il SEAL rappresenta probabilmente il più veloce algoritmo di crittografia attualmente disponibile. Le chiavi che utilizza richiedono diversi kilobyte di spazio, ma bastano solo 5 operazioni per byte per la generazione della keystream. Questo algoritmo è particolarmente appropriato

per le applicazioni di crittografia su dischi e in tutte le applicazioni nelle quali è necessario cifrare un blocco di dati con letture variabili dal centro. Il SEAL è registrato dall'IBM che ne detiene la licenza.

Algoritmi hash

MD2
L'MD2 è generalmente considerato un algoritmo morto. Sviluppato per lavorare sui processori a 8 bit, oggi, con l'avvento dei processori a 32/64 bit, risulta molto poco utilizzato.
Produce output di 128 bit. L'MD2 è differente nel design dall'MD4 e dall'MD5. Non ci sono attacchi conosciuti sulla sua versione completa.

MD4
L'MD4 è un algoritmo hash considerato sicuro, che utilizza varie tecniche crittografiche ed è considerato basilare per le funzioni hash. Vediamo brevemente come lavora. Per prima cosa il messaggio viene diviso in blocchi di 512 bit, successivamente, tramite una struttura iterativa di tipo Damgård/Merkle, il messaggio viene elaborato da una funzione di compressione sui blocchi di 512 bit per la generazione del valore hash. L'output ha una lunghezza di 128 bit. Hans Dobbertin ha sviluppato un attacco a questo algoritmo, in grado di generare collisioni in circa un minuto di tempo di calcolo su un normale PC. Una panoramica sulle sue specifiche progettuali si può trovare nei documenti rilasciati dall'RSA.

MD5
Mentre l'MD4 è stato sviluppato per la velocità di esecuzione, l'MD5 ha caratteristiche più robuste legate alla sicurezza. Infatti le stesse tecniche utilizzate per la generazione delle collisioni sull'MD4, nell'MD5 impiegano tempi maggiori dell'ordine di seicento volte. In particolare Hans Dobbertin ha dimostrato che con l'MD5, con un normale PC, occorrono circa 10 ore per trovare collisioni. L'MD5, come l'MD4, produce output di 128 bit.

RIPEMD
RIPEMD e gli algoritmi successivi sono stati sviluppati dal progetto European RIPE. Gli autori hanno scoperto collisioni per una sua versione, ristrette a due casi. Questi attacchi valgono anche per l'MD4 e l'MD5. L'algoritmo originale RIPEMD è stato successivamente modificato e migliorato nel RIPEMD-60, che utilizza output di 160 bit.

SHA1
È stato sviluppato dall'NSA su richiesta del NIST, come parte dello Standard Secure Hash (SHS). L'algoritmo è tecnicamente simile all'MD4. L'originale, conosciuto con il nome di SHA, è stato modificato dall'NSA per proteggerlo da ulteriori attacchi; il nuovo algoritmo è stato chiamato SHA1. Produce output di 160 bit digest, più che sufficiente per prevenire attacchi del tipo "birthday", dove due messaggi differenti sono utilizzati per la generazione della stessa funzione hash.

Snefru
È una funzione hash sviluppata da Ralph Merkle, l'ideatore degli algoritmi Khufu e Khafre. Una sua versione, la 2-round, è stata violata da Eli Biham. Sono in palio 1000 dollari per chi riesce a violare la versione 4-round Snefru. Le ultime versioni generano valori hash di 128 e 256 bit.

Tiger
È un nuovo algoritmo hash sviluppato da Ross Anderson ed Eli Biham. Sviluppato per lavorare su processori a 64 bit come il Digital Alpha, a differenza dell'MD4, non utilizza istruzioni di rotazione per la generazione del valore di hash. Per questioni di compatibilità con altri algoritmi hash, il Tiger consente di generare output a 128, 160 o 192 bit.

Quando la crittografia serve a poco

Siete seduti davanti al terminale e state scrivendo un documento riservato per la vostra azienda. Dopo aver salvato il lavoro decidete

di crittografarlo per nasconderlo a occhi indiscreti, ma ormai è troppo tardi! Il vostro documento potrebbe essere già stato copiato e diffuso alla concorrenza, mettendo a rischio il futuro dell'azienda e soprattutto il vostro posto di lavoro. Anche se questa ipotesi può sembrare ispirata a un film di fantascienza, la possibilità che una vicenda del genere accada nella vita di tutti i giorni è ormai molto alta. Per superare la sicurezza offerta da un sistema crittografico, è necessario riuscire a intervenire sul documento in chiaro prima che venga cifrato (quindi il "lucchetto" crittografico viene saltato a priori). Per farlo, dal momento che la crittografia viene usata soprattutto in ambito informatico, è necessario riuscire a copiare, in qualche modo, i bit informativi del documento. Una delle tecniche più sofisticate consiste nel captare, con un'adeguata attrezzatura elettronica (le cui specifiche sono di pubblico dominio), le onde elettromagnetiche emesse dal computer nel quale sono contenuti i file riservati, e copiarli su un altro computer che si trova a centinaia di metri dal primo (figura A.4). Come già detto, questa non è fantascienza, Nella realtà ogni dispositivo elettrico, nel momento in cui viene alimentato, genera campi elettromagnetici che si propagano nell'etere, attraverso onde, con distanze che variano a seconda della potenza del dispositivo che le emette (in media un comune personal computer emette onde elettromagnetiche per circa 700 metri di distanza!). Per avere un esempio concreto delle interferenze tra apparati elettrici, potete avvicinare un cellulare al monitor del computer e vedere cosa succede sullo schermo. Noterete delle piccole linee orizzontali che disturbano il segnale del monitor. L'unica difficoltà tecnica che si riscontra nel captare onde elettromagnetiche è la ricostruzione del segnale captato. Ritornando all'esempio precedente, se si deve ricostruire un file captato attraverso onde elettromagnetiche, si devono conoscere a priori i tipi di segnali emessi dal dispositivo. Questo perché ogni circuito elettrico, anche se strutturalmente simile agli altri, genera segnali leggermente diversi. A prescindere da questa difficoltà (peraltro risolta in molti casi grazie a circuiti in grado di effettuare lo scanning dei segnali su più frequenze contemporaneamente), rimane il fatto che un estraneo, e quindi un potenziale nemico, può captare i segnali provenienti dal nostro computer, regi-

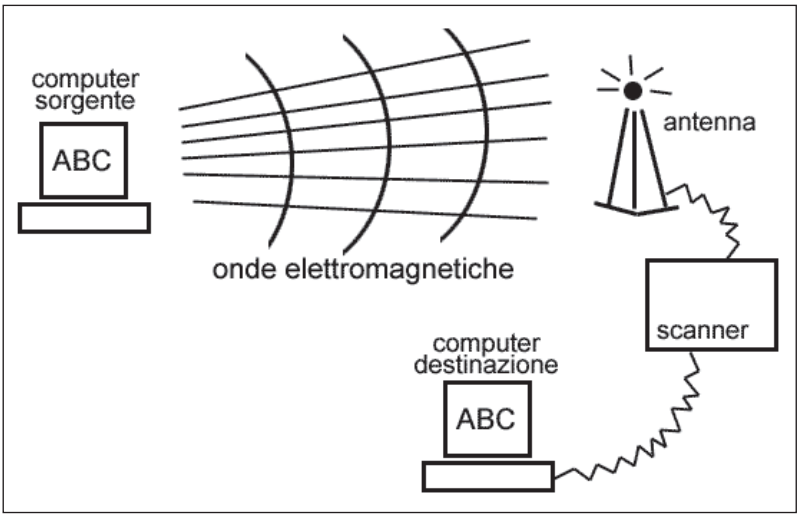

Figura A.4

strarli e successivamente decifrarli riuscendo a ricostruire i nostri documenti riservati. Oltre a questa tecnica, che sfrutta le onde elettromagnetiche, esistono anche altre soluzioni basate sull'ausilio di particolari programmi installati sul computer in cui risiedono i documenti riservati o su una stazione (server) con la quale il computer è collegato in rete[2].

Questi programmi, chiamati in gergo *sniffer*, consentono di analizzare il flusso dei dati di una rete e quindi sono in grado di copiare qualsiasi cosa venga loro indicata. In pratica possono essere programmati per registrare particolari sequenze di dati o addirittura per copiare fisicamente tutto il flusso di bit. Esistono anche sniffer che, invece di monitorare il flusso di reti, consentono di registrare i dati introdotti da unità di input come tastiere, mouse, scanner eccetera. In questo secondo caso, ad esempio, si potrebbero registrare tutte le pressioni della tastiera e ricostruire in questo modo le attività svolte dall'utente su una particolare macchina (molte password possono essere scoperte in questo modo prima che vengano cifrate). Queste ultime tecniche, a differenza dello scanning delle onde

elettromagnetiche, sono legate al mondo del software e quindi possono essere applicate solo se si ha libero accesso alla macchina da controllare (quindi fisicamente sul posto o tramite rete informatica). Nel caso delle onde elettromagnetiche, il grande vantaggio è costituito dal fatto che chiunque, a centinaia di metri di distanza dall'obiettivo, può copiare i dati presenti in un determinato computer. Come è possibile intuire, si tratta di una tecnica straordinaria e particolarmente pericolosa per agenzie di stato, settori industriali di punta eccetera, ossia in tutte quelle attività nelle quali si producono documenti riservati. Come proteggere i computer da questi possibili attacchi? La risposta è TEMPEST, almeno secondo l'*intelligence* americana.

TEMPEST, una tempesta di onde elettromagnetiche

Il progetto TEMPEST è uno standard di sicurezza dei dispositivi elettrici contro la propagazione delle onde elettromagnetiche. Nel 1950 il governo americano scoprì che era tecnicamente possibile captare le emanazioni di onde elettromagnetiche da un dispositivo e ricostruire, di conseguenza, il segnale originale. Questa scoperta suscitò subito un certo interesse nei settori di *intelligence* americani e, per combattere il problema, fu attivato il programma TEMPEST. Nato forse come acronimo di Transient Electromagnetic Pulse Emanation Standard[3], questo programma ha suscitato subito l'interesse delle grosse aziende del settore, che per ovvi motivi volevano partecipare alla realizzazione di uno standard per la difesa dall'emissione di onde elettromagnetiche (si parla di un giro d'affari di oltre 1000 miliardi di dollari). Nel corso degli anni vennero pubblicati molti documenti (la maggior parte riservati) nei quali il progetto TEMPEST veniva continuamente citato con allegati tecnici che accennavano alle possibili soluzioni.[4] Nel 1970 lo standard TEMPEST fu revisionato nel documento conosciuto con il nome di National Security Information Memorandum 5100 o NACSIM 5100. L'attuale standard è stato riconosciuto il 16 gennaio 1981 nel documento riservato NACSIM 5100A.

L'organo preposto al controllo dello standard TEMPEST è dal 1984 l'NSA, che si occupa di informare le varie agenzie di stato sui dispositivi

di sicurezza da installare nelle proprie sedi e attualmente è l'unico punto di riferimento per gli aspetti tecnici legati a questo standard. La preoccupazione principale nell'utilizzo delle tecniche legate al monitoraggio delle onde elettromagnetiche è incentrata sui monitor dei computer. Il fatto che si riesca a ricostruire ciò che appare sul video di un computer in cui sono memorizzati documenti riservati è la maggior preoccupazione dell'NSA. Nel momento in cui scriviamo, non risultano dichiarazioni ufficiali relative a casi legati allo standard TEMPEST. Resta il fatto che nella maggior parte dei casi la tecnologia per riuscire a captare segnali elettromagnetici e registrarli è ormai molto diffusa. Anche se gli apparati per questo tipo di intercettazioni sono estremamente costosi, nel circuito hacker sono circolati schemi tecnici per la costruzione di strumenti artigianali che consentono di ottenere risultati "interessanti". Gli elementi principali sono uno scanner multi-frequenza, una buona antenna e un sistema televisivo munito di videoregistratore VCR.
La sicurezza offerta dal sistema TEMPEST contro la diffusione di onde elettromagnetiche è basata su protezioni strutturali dei circuiti elettrici tramite celle di Faraday[5]. I costi di realizzazione di un dispositivo con specifiche TEMPEST mediamente raddoppiano; per questo motivo sono stati analizzati i vantaggi che potrebbero derivare dalla schermatura di ogni singolo PC presente in un edificio, invece dell'intero edificio (nella figura A.5, prelevata dal documento ufficiale Electromagnetic Pulse (EMP) And Tempest Protection For Facilities del corpo degli ingegneri militari degli USA sono riportate le specifiche per una schermatura totale di un luogo contenente documenti elettronici riservati, praticamente nessun segnale può uscire da quella stanza!). Spesso è stata preferita la seconda alternativa e sono numerosi gli edifici di governo americano che hanno adottato la schermatura totale della struttura ospitante invece di concentrarsi sulle singole stazioni di lavoro. In ambito architettonico sono state utilizzate diverse soluzioni per la schermatura dei segnali e in questo caso i costi derivanti per la schermatura si aggirano sui 50 dollari ogni 30 cm^2 (nella figura A.6, ad esempio, si possono notare alcuni particolari di realizzazione per la schermatura interna/esterna dei cavi elettrici). Nel mondo commerciale lo standard

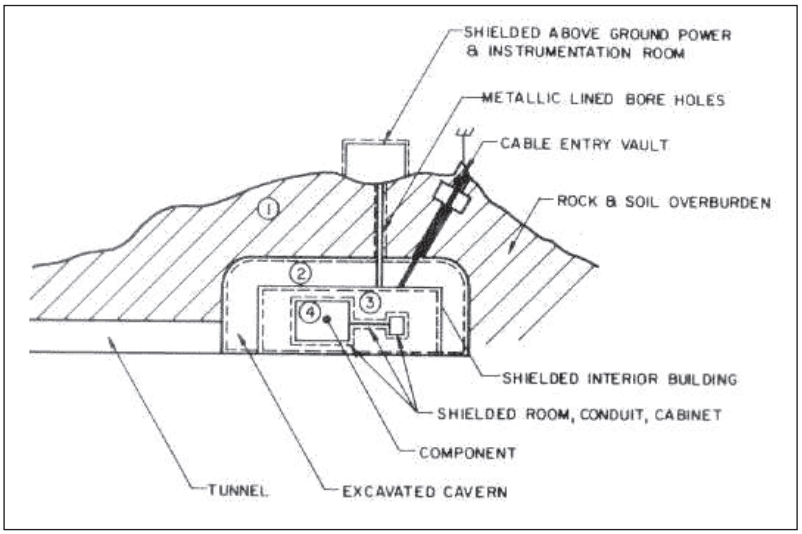

Figura A.5

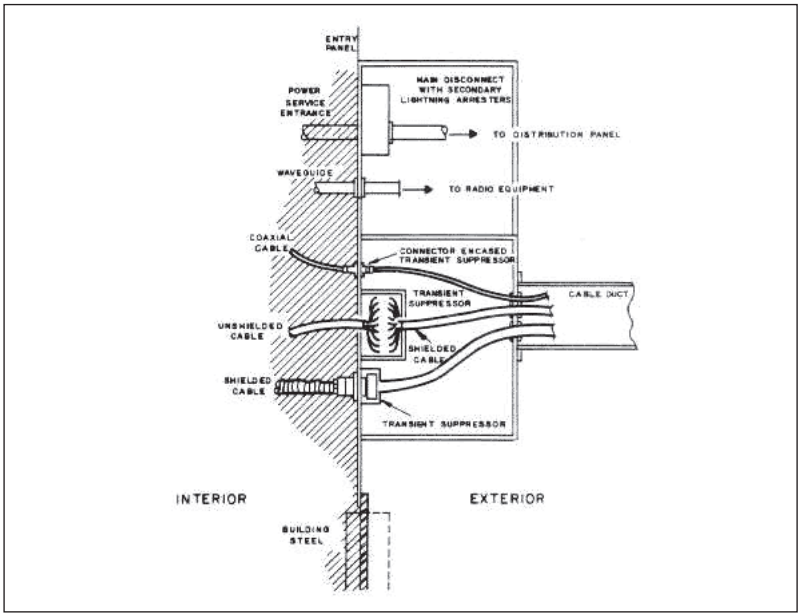

Figura A.6

TEMPEST è stato implementato solo in rarissimi casi, ad esempio l'IBM ha prodotto un case speciale per personal computer con un costo aggiuntivo di 35 dollari, chiamato EMR XT SYSTEM UNIT, modello 4455 1.

TEMPEST: l'unico rimedio?

Certamente lo standard TEMPEST non è l'unico in circolazione: esistono standard efficienti con costi ridotti e gradi di protezione paragonabili al TEMPEST, ad esempio ZONE. Ma quando è necessario utilizzare una protezione di questo tipo? Certamente non per uso personale – non crediamo che il vostro computer sia a rischio, per quanto riguarda la radiazione di onde elettromagnetiche. Un ipotetico nemico potrebbe mettere le mani sui vostri dati in maniera più semplice e soprattutto più economica (ad esempio copiandoli mentre siete collegati in Internet oppure semplicemente entrando nel vostro appartamento). Non poche critiche sono state rivolte nel corso degli anni allo standard TEMPEST. Il motivo principale rimane quello economico. Non solo i costi di realizzazione sono troppo alti ma, trattandosi di uno standard, bisogna adattarsi alle sue specifiche, altrimenti si rimane esclusi dal gioco delle certificazioni. Studi indipendenti hanno dimostrato che, per proteggere una apparecchiatura dall'emissione di onde elettromagnetiche, non è necessario adottare tutte le specifiche presenti nello standard TEMPEST. Secondo i casi, si possono utilizzare sistemi più economici, mirati a schermare in particolare i circuiti elettrici maggiormente a rischio (come le CPU dei computer, nelle quali viaggiano tutte le istruzioni e quindi i comandi immessi dall'utente) o semplicemente a confondere i segnali emessi, senza limitarsi a bloccarli (quest'ultimo caso è stato dimostrato dal Prof. Erhart Moller dell'università di Aachen in Germania, che ha anche messo in pratica questa tecnica costruendo la parte di intercettazione su un segnale video di un terminale).

Keyboard sniffer

Avevamo già accennato alla tecnica dello sniffer per la copia dei dati che viaggiano attraverso reti informatiche o che vengono inseriti tramite dispositivi di input. Vediamo più in dettaglio una cate-

goria particolare di tali software, i cosiddetti keyboard sniffer, ossia i programmi che consentono di copiare in un file tutte le pressioni eseguite sulla tastiera. Fondamentalmente essi funzionano in questo modo: una volta installati, consentono di registrare su un file, opportunamente nascosto e a volte cifrato, tutte le pressioni dei tasti di un utente sprovveduto. In questo modo, come è facile intuire, si possono copiare con estrema facilità le password di sistema di programmi protetti, i dati personali per l'accesso a una rete a pagamento e così via. Tecnicamente un keyboard sniffer si pone tra la shell del sistema operativo e il kernel del sistema; in pratica a ogni pressione viene attivata una procedura che registra in maniera sequenziale il tasto premuto in un file prestabilito. Un aspetto importante di questo software è l'invisibilità; infatti un utente che non sia particolarmente smaliziato non dovrebbe accorgersi che lo sniffer sta lavorando. Si ottiene l'invisibilità in vari modi, secondo il sistema operativo prescelto per l'attacco. Ad esempio in ambiente DOS si potrebbero utilizzare le tecniche dei TSR fantasma o della riprogrammazione del vettore degli interrupt corrispondente alla chiamata di lettura da tastiera (l'Int 16/00 del BIOS). In ambiente Unix si potrebbe utilizzare un processo ben mascherato, magari in uno di sistema; oppure in ambiente Windows lo si potrebbe inserire in un virtual device driver (VxD). In commercio, ma soprattutto nel mondo degli smanettoni o hacker, si possono trovare diversi esempi di keyboard sniffer per i vari sistemi operativi utilizzati. Vediamone due esempi, uno per DOS chiamato KeyTrap, e uno per sistemi Windows chiamato KeyLogger.

DOS: KeyTrap

Keytrap è un programma di pubblico dominio per sistemi DOS, distribuito con i sorgenti in linguaggio C e Assembly 80x86 e giunto alla versione 3.0. Una volta eseguito, si installa in memoria e vi rimane residente (TSR) fino a quando non si spegne il computer. Si basa sul principio degli interrupt 21h del DOS. In pratica aggiunge del codice ai normali servizi di lettura tasti del sistema, scrivendo ogni tasto premuto in un file prestabilito. Il programma funziona

anche se accidentalmente il file contenente i tasti premuti (log file) viene cancellato durante l'esecuzione. L'esecuzione è molto semplice e, dal momento che sono distribuiti anche i sorgenti, risulta possibile visionare direttamente il codice per non avere sorprese[6]. Keytrap può essere scaricato direttamente da Internet, l'indirizzo è `www.mhv.net/~dcypher/keytrap.html` (volendo si può contattare l'autore per chiedere spiegazioni particolari, all'indirizzo e-mail `dcypher@mhv.net`).

Windows: KeyLogger

KeyLogger, realizzato dalla società americana Amecisco, a differenza di KeyTrap non è di pubblico dominio, ma è un software shareware. Disponibile in due versioni, una chiamata KeyLogger97 e l'altra KeyLogger Stealth, che differiscono nelle prestazioni, riesce a registrare la pressione dei tasti utilizzando diverse opzioni per l'intercettazione e scaricando le informazioni su un file prestabilito opportunamente cifrato attraverso una tecnica di scrambler (ossia di confusione dei caratteri). Una volta installato, il programma rimane nascosto e non risulta visibile nella barra degli strumenti del desktop. Per rendersi invisibile KeyLogger utilizza la tecnica dei virtual device driver (vedi figura A.7) ponendosi in esecuzione e in continuo ascolto della tastiera tra le applicazioni e la GUI di Windows. Non crea nessun collegamento nel desktop e non provoca l'inserimento di alcun nel menu Avvio di Windows. Per attivarlo, è necessario riavviare il sistema o eseguire direttamente l'applicazione nella directory prescelta per l'installazione (file Ik.exe).

Quando il programma viene eseguito, sono intercettati tutti i tasti premuti sulla tastiera e registrati in un file cifrato chiamato Ik.dat. Per poterne leggere il contenuto, sarà necessario eseguire il programma dat2txt.exe, presente nella stessa directory di KeyLogger. Eseguendo questo programma, il file Ik.dat verrà decifrato nel file Ik.txt, leggibile attraverso un comune editor (tipo il Blocco Note). Il programma, in una versione demo, può essere scaricato liberamente da Internet all'indirizzo `www.amecisco.com`. La versione demo naturalmente è limitata in alcune funzioni: ad esempio intercetta solo 500 caratteri e non intercetta le sessioni di login e le password

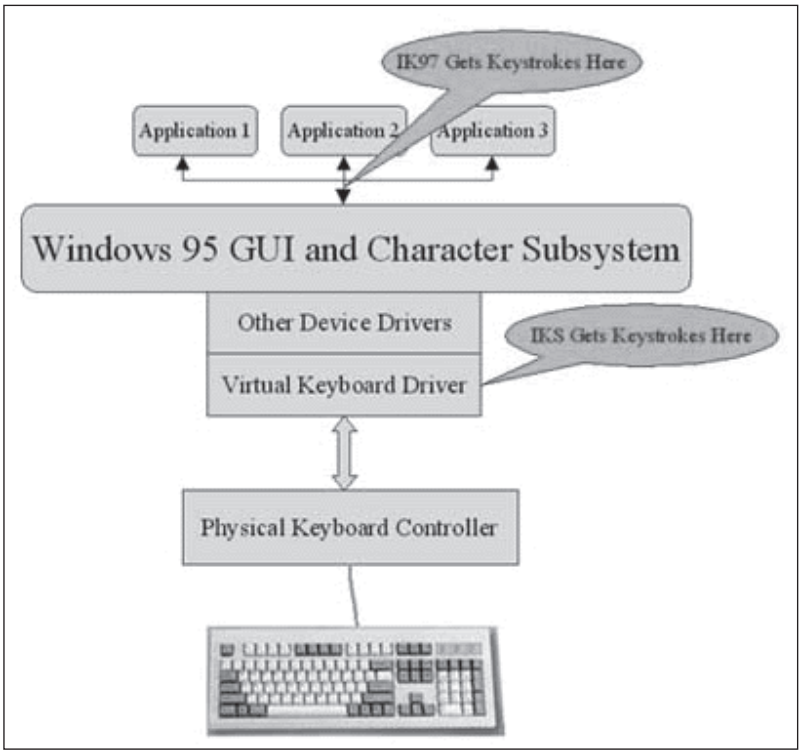

Figura A.7

di sistema di Windows. La versione registrata di Keylogger costa 29 dollari, mentre Keylogger Stealth ne costa 79. Il difetto di questo programma sta nel fatto che un utente, usando la combinazione di tasti Ctrl+Alt+Canc, potrebbe scoprirne l'esistenza tramite la riga Ik, chiedersi "...cosa sarà mai questo processo?" e quindi terminare l'applicazione, eliminando il problema almeno fino al prossimo reboot.

Scrambler telefonici

Quando utilizziamo il telefono per comunicazioni riservate, in effetti non abbiamo nessun tipo di protezione. Anche se nei moderni sistemi di comunicazione cellulare si utilizzano tecniche crittografiche per la protezione dei dati, abbiamo visto come esse risultino poco

efficaci e ormai superate (vedi il paragrafo “Clonazione dei GSM” al capitolo 10). Come garantire una privacy efficace utilizzando sistemi alla portata di tutti? Una prima risposta la si può trovare nell’utilizzo dei cosiddetti scrambler telefonici. Si tratta di dispositivi in grado di alterare un segnale elettrico, per mascherarlo e quindi proteggerlo durante un percorso di comunicazione. Una volta “alterato” un segnale, si dovrà utilizzare un altro dispositivo scrambler per poterlo ricostruire e di conseguenza decifrarlo all’atto della ricezione del segnale. Ovviamente i dispositivi scrambler utilizzati nella fase di cifratura e decifrazione devono operare allo stesso modo e quindi dovranno condividere una stessa “password”, per lavorare in maniera sicura. A livello teorico questi sistemi utilizzano un tipo di crittografia simmetrica, poiché usano una sola chiave per la cifratura e la decifratura delle telefonate. I primi modelli di scrambler telefonici impiegavano sistemi molto semplici per l’alterazione del segnale. Si trattava di piccoli “trucchi” elettronici legati alla sovrapposizione di forme d’onda particolari, a tagli di frequenza eccetera, che potevano essere ricostruiti “artigianalmente” tramite l’utilizzo di oscilloscopi e generatori di forme d’onda. Le password erano costituite dai valori numerici relativi alle frequenze di taglio, al tipo di forma d’onda alterante e così via, e quindi l’insieme delle possibili chiavi risultava abbastanza limitato (di solito non si superavano le 5000 possibilità). Con il passare del tempo questi dispositivi sono stati rimpiazzati da scrambler più efficaci dal punto di vista tecnico e soprattutto crittografico. Un esempio moderno è l’AT&T Security Telephone Device 3600 (figura A.8), uno scrambler che utilizza un sistema di negoziazione delle chiavi e un algoritmo proprietario della ditta americana AT&T per la fase di cifratura. Questo dispositivo, dal costo di 1000 dollari circa, garantisce la privacy durante la comunicazione attraverso l’utilizzo di un algoritmo crittografico basato sulla conoscenza di una chiave di sessione generata al momento dell’attivazione del dispositivo (e quindi personalizzata solo per quella particolare telefonata) e comunicata all’altro scrambler del destinatario. Esistono anche scrambler telefonici per le comunicazioni via fax: un esempio, sempre della stessa compagnia americana AT&T, è il modello 3700/3710 che consente di comunicare in maniera sicura at-

Figura A.8

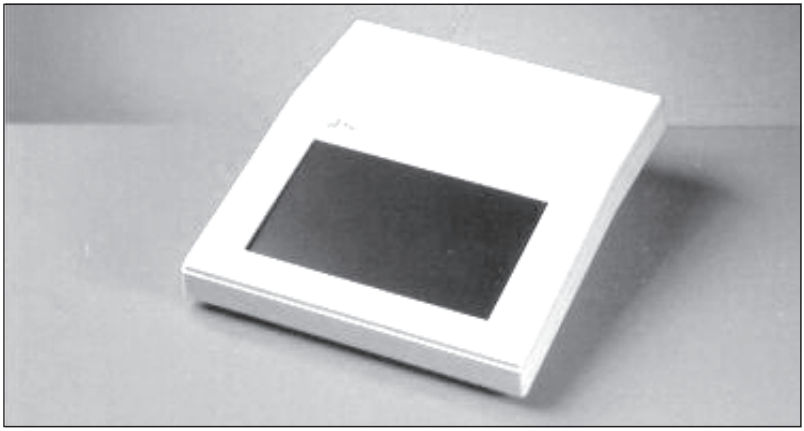

Figura A.9

traverso un algoritmo proprietario e una gestione avanzata delle chiavi, tramite la generazione di numeri casuali (figura A.9). Il costo di questi scrambler per fax è decisamente più elevato, si superano abbondantemente i 2000 dollari.
Esistono anche scrambler più economici di quelli appena presentati, che a livello crittografico utilizzano sistemi meno sofisticati. Uno di essi è l'Enigma 100 (figura A.10) basato sull'utilizzo digitale di

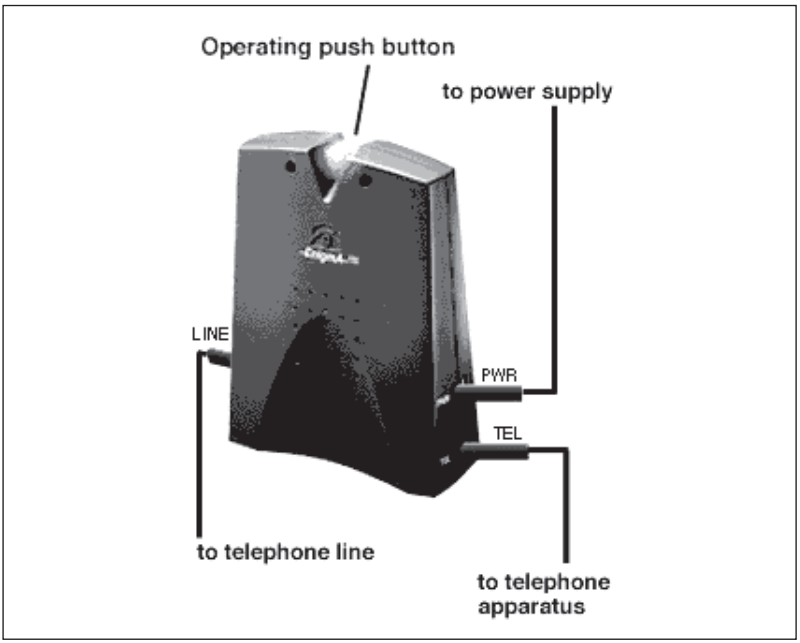

Figura A.10

un DSP (Digital Signal Processing) per la protezione e l'elaborazione dei segnali elettrici. Esistono anche scrambler per i cellulari; in questo caso le tecniche utilizzate per la protezione del segnale sono molto più sofisticate, dovendo lavorare con dati digitalizzati e quindi con numeri facilmente "mascherabili" grazie alle tecniche matematiche della crittografia.
Anche se consentono di ottenere buoni risultati dal punto di vista della sicurezza, i sistemi descritti non sono proprio alla portata di tutti, visti i costi e le esigenze pratiche legate a un utilizzo hardware di due dispositivi posizionati ai capi di una comunicazione telefonica. Esistono però tecniche altrettanto sofisticate, se non di più, e molto più pratiche e soprattutto economiche: gli scrambler telefonici software. Dal momento che ormai la maggior parte dei computer è dotata di una scheda audio e di un modem, perché non combinare le due cose per alterare le telefonate con questi dispositivi?

Vediamo come. Per mascherare una telefonata tramite l'utilizzo di un normale personal computer, è necessario in qualche modo digitalizzare la voce, modificarla con un algoritmo crittografico e inviarla tramite modem nella linea telefonica. La digitalizzazione e la modifica del segnale che rappresenta la voce vengono effettuate da un software particolare; ne esistono diversi in circolazione (alcuni anche disponibili gratuitamente) che, tramite una scheda audio e un microfono, riescono a convertire la forma d'onda sonora in una lunga catena di numeri facilmente manipolabili da un algoritmo crittografico. Ad esempio, il programma freeware "Speak Freely" realizzato da John Walker e disponibile per molte piattaforme – tra le quali l'ultima versione 6.1 per Windows 95 – consente di proteggere le comunicazioni telefoniche utilizzando come mezzo di trasporto la rete Internet. In questo modo la comunicazione telefonica viene convertita in un normalissimo pacchetto TCP/IP che viaggia nella rete, per arrivare a destinazione e subire il processo di riconversione da digitale cifrato ad analogico in chiaro. Il programma utilizza l'algoritmo IDEA per le operazioni di cifratura e consente di chiamare qualsiasi utente sparso in Internet a patto di conoscere il suo indirizzo IP, ossia il numero identificativo che differenzia ogni utente che accede in rete. Con questo software, ad esempio, è possibile comunicare con un amico in maniera sicura ed economica, dal momento che la "telefonata" avviene utilizzando la rete e quindi i costi, in teoria, sono nulli (in realtà i costi reali sono identificati dal costo della telefonata, di solito urbana, effettuata per il collegamento al provider fornitore dell'accesso a Internet).
Esistono molti altri programmi che consentono di proteggere le conversazioni telefoniche attraverso l'utilizzo di Internet: ad esempio il PGPFone, realizzato dal team di Philip Zimmermann, l'autore del famoso PGP, o il Nautilus, un software anch'esso freeware giunto alla versione 1.7b per Windows e disponibile con i sorgenti. Ovviamente questi tipi di software sono ancora in fase sperimentale e, visti i problemi di traffico della rete, soprattutto nella situazione italiana, passerà ancora del tempo per poterli utilizzare al meglio, come in una normale conversazione telefonica. I ritardi di trasmissione e la scarsa qualità del segnale non consentono comunica-

zioni full duplex per la maggior parte degli utenti Internet: bisogna sapersi accontentare. “Le conversazioni telefoniche sono personali, sono private, e non sono affari di nessuno se non nostri. Potremmo pianificare una campagna politica, discutere di tasse o avere una relazione clandestina ... Di qualunque argomento trattino, non vogliamo che le nostre telefonate siano intercettate o ascoltate da qualcun altro. Non c’è nulla di male nel voler affermare il nostro diritto alla privacy” (Philip Zimmermann).

Note

1 Nel caso in cui il telefonino squilli il disturbo sul video dovrebbe accentuarsi; se non avete a disposizione un cellulare, potete provare con una radio o con un televisore, l’effetto dovrebbe essere lo stesso.
2 Ad esempio una comune LAN, Local Area Network, diffusa ormai in quasi tutti gli ambienti di lavoro.
3 Il governo americano nega qualsiasi spiegazione, affermando che il nome TEMPEST è casuale.
4 Inizialmente il progetto fu chiamato con la sigla NAG1A, successivamente FS222 e FS222A.
5 In pratica, il circuito viene schermato con una barriera di materiali speciali contro la diffusione dei segnali.
6 Questa politica della diffusione dei sorgenti negli ambienti hacker denota una totale trasparenza nella diffusione delle informazioni, una politica appoggiata da molte realtà nella rete, una tra tutte la Free Software Foundation, con il progetto GNU di Richard Stallman, www.gnu.org.