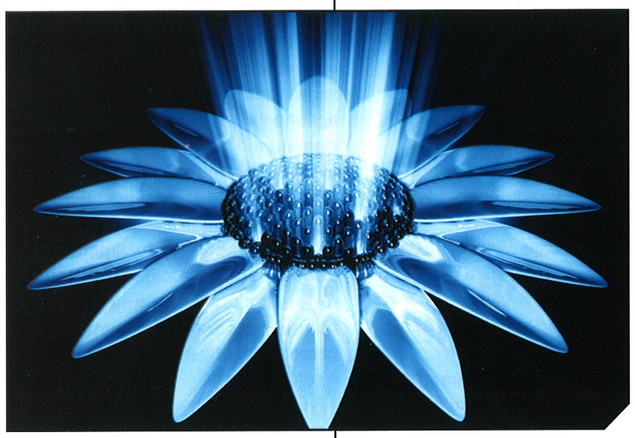
UNA NUOVA STAGIONE PER LA CRITTOGRAFIA

Sandro Fontana
Secure Edge S.r.l.
sfontana@secure-edge.com
s.fontana@computer.org
CISA, CISM, CISSP, L.A. BS7799
ACM, AICA, CLUSIT, IEEE, ISACA, ISSA, USENIX

In questi ultimi 24 mesi, sono stati presentati una serie di attacchi agli algoritmi di crittografia su cui basiamo la sicurezza dei nostri dati. Nessuno di questi attacchi è attualmente capace di minare l'infrastruttura di sicurezza comunemente utilizzata; tuttavia dobbiamo prendere atto che è il momento di iniziare a prepararci per un cambiamento. Il documento che segue riepiloga alcuni di questi fatti e mette in evidenza i limiti attuali degli algoritmi utilizzati, proponendo come alternativa quanto già valutato ed accettato dalle più affidabili organizzazioni internazionali.

Alcuni rumors su di una presunta vulnerabilità nello SHA-1, espressi nell'agosto del 2004 sono poi diventati notizia i primi mesi del 2005: in una università cinese due ricercatori[1] hanno trovato il modo di generare coppie di sequenze binarie che date in input allo SHA-1 generano lo stesso valore di hash. Questo attacco allo SHA-1 è oggi conosciuto come *Xiaoyun Wang's attack*. Alcuni post sulle mailing list ed alcuni articoli apparsi sulla rete hanno interpretato queste notizie come "*La fine della fima digitale!*". Al contrario altri hanno rifiutato di credere che ci fosse qualcosa di vero su quanto veniva riferito sui ricercatori cinesi e comunque non hanno accettato di vedere messo in crisi il valore della firma digitale in quanto "*gli algoritmi di crittografia asimmetrica sono sicuri per definizione!*".



Come spesso succede, in queste affermazioni c'è del vero e del falso; vediamo di ricapitolare lo stato dell'Arte sulle tre classi di algoritmi crittografici: hash, crittografia simmetrica, crittografia asimmetrica.

Algoritmi di hash

Lo Standard Hash Algorithm Version 1 o SHA-1 è relativamemte conosciuto anche tra i non specialisti di crittografia, in quanto rappresenta un componente fondamentale utilizzato nel processo di firma digitale.

Quando viene firmato un messaggio (un documento elettronico o più in generale un qualsiasi file), ciò che viene realmente firmato, cioè codificato con la chiave privata del firmatario, è l'*hash* del messaggio, cioè la sua impronta univoca e di lunghezza fissa. Oggi l'algoritmo di hash universalmente riconosciuto ed utilizzato è lo SHA-1. La scelta di codificare l'hash di un file al posto del messaggio stesso deriva da un problema prestazionale: codificare anche poche decine di kilobyte utilizzando gli algoritmi di crittografia asimmetrica è, computazionalmente parlando, un grosso impegno per un computer, specie se quest'ultimo è rappresentato dal chip di una smart card.

Si preferisce quindi firmare l'hash del documento, cioè una sequenza binaria di lunghezza fissa, che nel caso di utilizzo dello SHA-1 è di soli 160 bit.

Naturalmente perchè questa scelta abbia senso, ci sono due caratteristiche importanti che un algoritmo di hash deve rispettare:

I° l'algoritmo deve essere mono-direzionale (*one-way* o *pre-image resistance*); questo sta a significare che preso un qualsiasi[2] messaggio in input (da qui in avanti ***m***), calcolarne il suo valore di hash è facile, ma contemporaneamente dato un valore di hash è *impossibile* ricreare il valore originale ***m***. Inotre viene definito *2nd pre-image resistance*, il fatto che dato ***m*** ed il suo valore di hash, sia impossibile trovare un m_1 diverso da ***m*** tale che l'hash di m_1 sia uguale all'hash di ***m***.

II° l'algoritmo non deve generare collisioni (*collision-free* o *collision resistance*); questo sta a significare che anche se ci sono un numero infinito di messaggi in input che possono dare lo stesso valore di hash[3], non sarà mai possibile trovare anche solo due di questi messaggi.

È importante ricordare che i concetti di *impossibile* e *mai*, in crittografia, non sono assoluti; stanno piuttosto a significare che non si può raggiungere il risultato in questione, in un ammontare di tempo conveniente.

Un attacco allo SHA-1

Cosa è successo quindi in Cina? Sembrerebbe proprio che lo Xiaoyun Wang's attack riesca a violare la caratteristica II° di un algoritmo di hash; in Cina quindi, si riescono a generare coppie di input che sottoposte allo SHA-1 danno lo stesso risultato; naturalmente , ci si aspetta che a breve altri specialisti di crittografia nel mondo riescano ad emulare questo attacco e magari iniziare a gettare le basi per un attacco anche alla caratteristica I°.

Questo significa che una firma digitale può essere violata?

Attualmente no, stiamo parlando di tutt'altra cosa. Trovare una coppia di input che restituisca il medesimo valore di hash —quanto realizzato dall'attacco alla caratteristica II°— è completamente differente dal trovare un input che corrisponde ad un dato valore di hash, cosa che effettivamente metterebbe in crisi il concetto di firma digitale essendo un attacco alla caratteristica I°.

Tutto bene allora?

Ovviamente no. Se pure l'attacco attuale, cioè la perdita dell'attributo di *collision-free* non implica la facilità di aggirare anche gli attributi di pre-image e second pre-image resistance, è pur vero che una delle caratteristiche fondamentali di un algoritmo di hash è stata violata; questo rende tutta la famiglia degli SHA (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)[4] non più affidabile, proprio come successe alcuni anni fà ad un'altro algoritmo di hash: l'MD5.

Se ne deduce che è arrivato il momento di pensare a migrare verso un'altro algoritmo di hash.

Un nuovo algoritmo di hash

Sostituire un algoritmo di crittografia e specialmente un algoritmo di hash, non è un compito facile né da prendere alla leggera: non molte istituzioni potrebbero incaricarsi di questo compito.

Il National Institute of Standard and Technology (NIST) degli USA si è guadagnato una grande reputazione in questo campo quando dichiarò non più affidabile il Data Encryption Algorithm (DEA), sostituendolo temporaneamente con il TDEA (Triple DEA) organizzando poi una gara a livello mondiale per decidere il suo successore. La gara occupò i migliori esperti di crittografia di tutto il mondo per circa quattro anni e —come ammette lo stesso Bruce Schneier—fu un lavoro duro che insegnò a tutti loro moltissimo.

Oggi è di nuovo il NIST il primo attore delle scene, annunciando una nuova gara internazionale per cercare il prossimo algoritmo di hash: le proposte di partecipazione verranno vagliate fino a tutto il 2008, quindi si ipotizzano una serie di studi, ana-

1] Yiqun Lisa Yin e Hongbo Yu del team del Prof. Xiaoyun Wang (Università di Shandong, Cina)
2] Di lunghezza massima 2^{64-1} bit
3] Avete letto bene, ma state pure tranquilli, tutto questo rientra nella logica di funzionamento degli algoritmi di hash. Per ogni valore di hash possibile, ad esempio usando SHA-1 per ognuno dei 2^{160} valori che possono essere espressi da questa funzione, ci sono infiniti input da cui partire. E' facile da capire: provate a mappare il concetto di INFINITO NUMERO DI INPUT in un numero come 2^{160} che è indubbiamente un numero grande, ma vale praticamente zero in confronto ad infinito!
4] Le ultime quattro varianti sono definite genericamente SHA-2

lisi ed incontri, fino ad arrivare alla definizione del nuovo standard entro la fine del 2011.
Tutti i più grandi crittografi hanno ammesso che il disegno del nuovo algoritmo di hash sarà un lavoro molto più difficile di quanto non sia stato quello di sviluppare un nuovo algoritmo di crittografia simmetrica: non c'è infatti molto know how sull'argomento e poche basi matematiche su cui appoggiarsi. Nel frattempo il consiglio unanime è di passare ad utilizzare il fratello maggiore dello SHA-1, cioè lo SHA-256.
Ricapitolando: dal 26 novembre 2001 abbiamo il nuovo standard di crittografia simmetrica, l'Advanced Encryption Standard (AES) ed entro il 2011 avremo un nuovo algoritmo di hash; non resta quindi che dare un'occhiata alla terza classe di algoritmi crittografici.

Gli algoritmi di crittografia asimmetrica

Questi algoritmi, nati inizialmente per risolvere il problema della distribuzione delle chiavi di crittografia simmetrica, si sono rivelati perfetti per autenticare il mittente di un messaggio: in parole povere questi algoritmi permettono la generazione della firma digitale[5].
Quello che ci interessa puntualizzare in questi algoritmi è il *cryptoperiod* delle attuali chiavi di firma digitale in uso.
Per *cryptoperiod* si intende il periodo di tempo durante il quale una specifica chiave è autorizzata ad essere utilizzata da soggetti legittimi o durante il quale un certo sistema rimarrà attivo; in breve: per quanto ancora sarà considerata robusta la chiave che si sta usando?
Nella pubblicazione SP 800-57 Part-1 del NIST, sono presenti alcune tabelle interessanti.
La *Table 2: Comparable strengths* mette in relazione la lunghezza delle chiavi usate da differenti algoritmi di crittografia simmetrica ed asimmetrica a parità di robustezza; in particolare risulta evidente che la robustezza delle attuali chiavi RSA da 1024 bit viene equiparata a quella di una chiave simmetrica da 80 bit. Quest'ultimo valore non è da tempo accettato come lunghezza delle chiavi di crittografia simmetrica: in genere la lunghezza reputata minima per questi algoritmi è 128 bit. Ebbene, affinché una chiave RSA possa avere una robustezza equivalente a quella di una chiave simmetrica da 128 bit, questa chiave RSA dovrebbe essere lunga almeno 3072 bit!
Il meglio però deve ancora arrivare: nella *Table 4: Recommended algorithms and minimum key sizes* il NIST dichiara esausto il cryptoperiod delle chiavi simmetriche da 80 bit e quindi di quelle RSA da 1024 bit entro il 2010!
Questo significa che se le informazioni codificate e/o firmate —ieri, oggi o domani— devono poter mantenere la propria sicurezza (integrità, riservatezza, non ripudio) oltre il 2010, bisogna accertarsi che la lunghezza delle chiavi utilizzate sia superiore agli 80 bit per la crittografia simmetrica e 1024 per la crittografia asimmetrica.
In particolare per informazioni su cui dare garanzia di sicurezza fino al 2030, bisognerà usare chiavi simmetriche almeno da 112 bit e chiavi RSA almeno da 2048 bit.
Se si vuole superare l'anno 2030 ed avere ancora la garanzia di sicurezza le chiavi simmetriche non possono essere più piccole di 128 bit mentre quelle RSA non dovranno risultare inferiori a 3072 bit.

Ebbene, si scelgano chiavi più lunghe !

Purtroppo le cose non sono così semplici. Il tempo di calcolo necessario alle chiavi più grandi non incrementa linearmente.
Con gli algoritmi RSA, il tempo necessario a calcolare una firma aumenta in modo proporzionale al cubo della lunghezza della chiave: firmare con una chiave RSA da 2048 bit occupa circa 8 volte il tempo di una firma RSA da 1024; se volessimo una firma con una robustezza equivalente a quella di un AES-256, dovremmo usare una chiave RSA da 15360 bit e per firmare ci metteremmo circa 3375 volte il tempo oggi necessario ad una firma RSA-1024: diciamo mediamente, circa 200 secondi!
Il problema diventa ancora più grande con le piccole device: PDA, Smartphone e Smart Card già attualmente, mal si adattano a gestire grandi chiavi RSA.
Inoltre anche sull'algoritmo RSA è stato presentato un attacco capace di svelare il contenuto di messaggi SSL/TLS quando uno dei parametri della chiave pubblica (l'*esponente pubblico*, un numero primo >2 preso a caso) ha come valore 3.
Anche se alcuni prodotti, ad esempio OpenSSL, usano come esponente pubblico il valore "65537", l'alternativa più comune è "3". Naturalmente il consiglio unanime è di smettere di utilizzare chiavi RSA che abbiano come esponente pubblico il valore "3".

Tutto è perduto?

Relax. Tutto ciò, anche se è preoccupante, fa parte della vita stessa della crittografia: è già avvenuto in passato e con tutta probabilità continuerà a succedere nel futuro. Anche in questo caso gli standard che, nel bene e nel male, gli USA impongono/offrono al resto del mondo, aiuteranno a definire un corretto ambito che gli specialisti di crittografia, a

5 Per Firma Digitale, si intente un qualsiasi processo di codifica basata sulla chiave privata di una coppia di chiavi asimmetriche, indipendentemente dall'uso di certificati qualificati o meno; ad esempio per intenderci, PGP ed il suo web-of-trust , essendo basato su crittografia asimmetrica, permette di generare una firma digitale.

livello mondiale, utilizzeranno per definire il prossimo futuro di questa scienza.

Il Federal Information Processing Standard (FIPS) 186-3, abilita tre tipi di algoritmi di crittografia asimmetrica per generare una firma digitale: il Digital Signature Algorithm (DSA), l'RSA[6] e l'Elliptic Curve Digital Signature Algorithm[7] (ECDSA).

A fronte dei problemi evidenziati nell'utilizzo futuro degli algoritmi RSA, prenderemo in considerazione l'Elliptic Curve Digital Signature Algorithm (ECDSA).

Una curva ellittica è una entità geometrica/algebrica, cioè è una curva piana derivata da una equazione del tipo $x^3 - y^2 + a\,x + b = 0$.

La loro applicazione in crittografia[8] avviene in un particolare ambito matematico, quello dei campi finiti, spesso indicati come GF[9] (la matematica dei numeri interi modulo un numero primo).

Nei GF, quando vengono usate particolari operazioni dette moltiplicazioni scalari, diventa applicabile il concetto del logaritmo discreto, che è un parente prossimo del problema della fattorizzazione dei numeri interi, base degli algoritmi RSA.

Il logaritmo discreto nei gruppi delle curve ellittiche, ovvero l'Elliptic Curve Discrete Logarithm Problem (ECDLP), rende queste adeguate alla costruzione di un cryptosystem.

In realtà quella delle curve ellittiche è una famiglia di algoritmi che variano in funzione di complesse scelte matematiche che in questo momento non ci interessano e che non sarei comunque in grado di spiegare (ne di capire) fino in fondo[10].

La cosa interessante nella Elliptic Curve Cryptography (ECC) è che a parità di sicurezza, i suoi algoritmi necessitano di chiavi molto più corte e questo si traduce in un grande aumento delle prestazioni. Chiavi più piccole significano quindi minore impegno di calcolo (più velocità), meno consumo di energia, e risparmi in termini di memoria e di banda utilizzate

Solo a titolo di esempio, si ricorda che una chiave RSA-1024, l'attuale

lunghezza di chiave RSA utilizzata per una firma digitale, ha come corrispettivo un ECC-160, cioè una chiave da 160bit: a livello di performance il rapporto tra i due algoritmi e le loro corrispondenti chiavi è di 1:4 in favore dell'ECC.

Le performance di ECC 224, equivalente di un RSA 2048, sono mediamente 9 volte più alte di quest'ultimo ed i rapporto può raggiungere le 14 volte su un grande server con 64 processori.

In ultimo due punti importanti:

a) va valutato che il rapporto tra i due tipi di algoritmi, sulle piccole device può raggiungere anche i due ordini di grandezza.
b) l'uso di firme di ridotte dimesioni, bene si adatterà al sempre più pervasivo utilizzo degli RFID e al nascente campo dei messaggi sicuri (ticket, ricevute ...) via SMS / MMS.

Ma la Elliptic Curve Cryptography è sicura?

Lo studio dell'ECC è più giovane di quello RSA, ma ha comunque più di 20 anni; il suo uso è stato standardizzato dall'ISO e dallo IETF e negli USA anche dall'ANSI e del NIST.

La sua implementazione è un poco più complessa di quella RSA, ma anche in questo gli istituti di standardizzazione USA ed EU daranno un grande aiuto.

Di certo c'è solo che ci sarà da rimboccarsi le maniche e magari togliersi tutta la camicia:

- ci sarà bisogno di un nuovo standard di Application Programming Interface (API) per l'ECC, accetta-

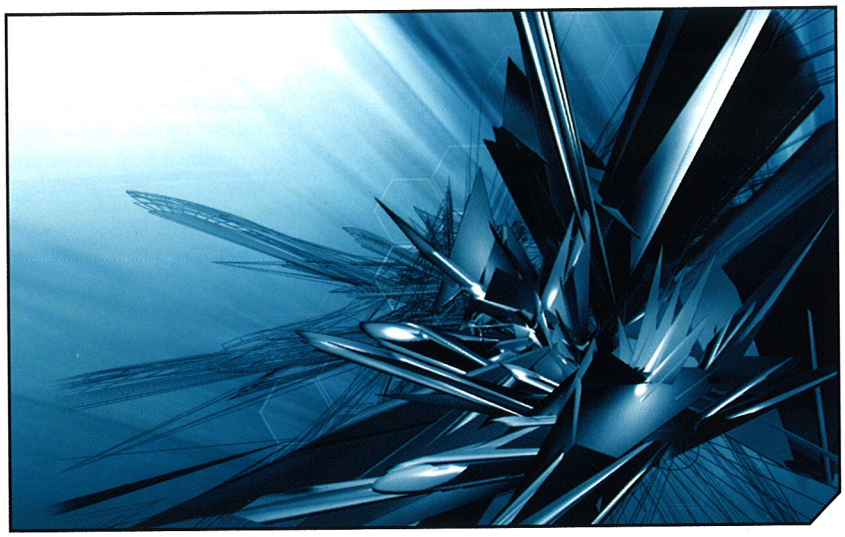
6 Dai nomi di chi lo ha sviluppato: Rivest, Shamir e Adelman

7 L'ECC fu inventato da Neil Koblitz e Victor Miller nel 1985, otto anni dopo l'algoritmo RSA

8 L'applicazione delle curve ellittiche alla crittografia è stata proposta contemporaneamente ed indipendentemente nel 1985 da Neal Koblitz dalla University of Washington e da Victor Miller ricercatore nel laboratorio IBM di Yorktown Heights

9 Acronimo per Galois field in onore del matematico francese Évariste Galois che per primo li definì.

10 Chi volesse tentare un'approfondimento, può trovare pane per i suoi denti nelle pubblicazioni del NIST e dell'ETSI riportate in bibliografia.



to almeno quanto l'attuale PKCS#11[11];

- pochi Hardware Security Module (HSM), Smart Card comprese, supportano l'ECC; questi dovranno quindi essere aggiornati o nella migliore delle ipotesi ricertificati secondo gli standard emanati nel "*CEN/ISSS workshop agreement*" (CWA 14167-1, CWA 14167-2, CWA 14169)[12], in pratica dei protection profile da usare nella certificazione Common Criteria;
- le Certification Authority dovranno rivedere le loro policy (CP/CPS) per includere gli stringenti requirement (molto diversi rispetto a quanto necessario con gli algoritmi RSA) per la generazione dei parametri e delle coppie di chiavi in ambiente ECC;
- ci sarà sicuramente una lotta nel mercato per essere i nuovi leader: Certicom, RSA, ... magari la Microsoft (tanto si mette dappertutto);

A noi piccoli utenti o nella migliore delle ipoptesi, smart-user dei sistemi di crittografia, non resta che stare a vedere e cercare di capire le implicazioni del tutto.

È probabile comunque che ciò che alla fine emergerà, sarà la cosidetta SUITE-B[13] o qualcosa di molto vicino a questa; attualmente questa ideale suite di crittografia risulta composta da:

- Symmetric Cryptography: AES FIPS 197 (da 128 a 256 bit)
- Digital Signature: Elliptic Curve Digital Signature Algorithm - FIPS 186-2 (curve ellittiche con modulo primo da 256 a 384 bit)
- Key Exchange: Elliptic Curve Diffie-Hellman or Elliptic Curve MQV (curve ellittiche con modulo primo da 256 a 384 bit)
- Hashing: Secure Hash Algorithm - FIPS 180-2 (solo SHA-256 o SHA-384)

L'insieme di questi strumenti sembra rappresentare quanto necessario a traghettarci per almeno tre/quattro decadi.

Ma la Elliptic Curve Cryptography è legale in Italia?

Ma che domande fate !
Magari adesso tirate pure in ballo la CIE! All'impossibile possiamo cercare qualche risposta, ma per i miracoli non siamo ancora attrezzati.

11 esiste già il PKCS#13, ma RSA lo dichiara "under develompment" dal 1998; daltronde fino ad oggi non è stato interesse di RSA di evolvere nella ECC

12E probabilmente gli stessi CWA dovranno essere revisionati/aggiornati alla luce dei nuovi eventi

13 Annunciata come tale dalla National 13 Security Agency (NSA) durante la 2005 RSA Conference.

Bibliografia

Near-Collisions of SHA-0
Eli Biham, Rafi Chen - International Association for Cryptologic Research - 22 Jun 2004
http://eprint.iacr.org/2004/146

SHA-1 Broken
Bruce Schneier - February 15, 2005
www.schneier.com/blog/archives/2005/02/sha1_broken.html

Fact Sheet NSA Suite B Cryptography
NSA
http://www.nsa.gov/ia/industry/crypto_suite_b.cfm

Recommendation for Key Management – Part 1: General
NIST nella pubblicazione SP-800-57 - August 2005
http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part1.pdf

Assurance for Digital Signature Applications
NIST nella pubblicazione SP-800-89 - November2006
http://csrc.nist.gov/publications/nistpubs/800-89/SP-800-89_November2006.pdf

Electronic Signatures and Infrastructures (ESI);Algorithms and Parameters for Secure Electronic Signatures;Part 1: Hash functions and asymmetric algorithms
ETSI nella pubblicazione ETSI TS 102 176-1 V1.2.1 (2005-07)
http://www.etsi.org/services_products/freestandard/home.htm

Advanced Encryption Standard (AES)
NIST nella pubblicazione FIPS 197
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Elliptic Curve Digital Signature Algorithm
NIST nella pubblicazione FIPS 186-2
http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf

Elliptic Curve Diffie-Hellman or Elliptic Curve MQV
Draft NIST Special Publication 800-56
http://csrc.nist.gov/CryptoToolkit/kms/keyschemes-Jan03.pdf

Secure Hash Algorithm
NIST nella pubblicazione FIPS 180-2
http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf

La normativa sulla firma elettronica
CNIPA
www.cnipa.gov.it/site/_files/minigrafia14_FirmaElettronica_.pdf