DigitCult | Scientific Journal on Digital Cultures

Crittografia: tra successi e fallimenti

Andrea Visconti
Dipartimento di Informatica
Università degli Studi di Milano
via Celoria 18, Milano, Italy

Abstract

La crittografia è una disciplina scientifica le cui origini possono essere datate migliaia di anni fa. Oggigiorno, nell'era di Internet delle cose, degli oggetti intelligenti, dell'industria 4.0 e delle crittomonete, la crittografia è divenuta uno strumento essenziale per la protezione dei nostri dati. Milioni di persone la utilizzano quotidianamente e inconsapevolmente. Anche se dal punto di vista teorico gli algoritmi crittografici risultano essere perfettamente sicuri, ogni tanto alcune loro implementazioni vengono effrante. In questo articolo introdurremo i principali punti di forza e debolezza che i crittografi hanno osservato in oltre duemila anni di storia.

Cryptography: between Successes and Failures

Cryptography is a science whose origins can be dated many years ago. Nowadays, in the Era of Internet of Things, smart objects and cryptocurrencies, it has become an essential tool for the protection of our data. Millions of unwitting users play with crypto algorithms everyday. Although these algorithms are theoretically secure, sometime specific implementations are broken. In this manuscript, we will introduce the main strengths and weaknesses observed over two thousand years of history.

Published 30 December 2019

Correspondence should be addressed to Andrea Visconti, Dipartimento di Informatica, Università degli Studi di Milano. Email: andrea.visconti@unimi.it

DigitCult, Scientific Journal on Digital Cultures is an academic journal of international scope, peer-reviewed and open access, aiming to value international research and to present current debate on digital culture, technological innovation and social change. ISSN: 2531-5994. URL: http://www.digitcult.it

 doi:10.4399/97888255301552

Introduzione

La crittografia è una materia scientifica multidisciplinare che coinvolge matematici, informatici e ingegneri. All'interno della comunità crittografica, questi attori svolgono prevalentemente tre tipi di attività. C'è chi studia nuovi algoritmi crittografici, c'è chi li implementa in software e/o in hardware e c'è chi si occupa di valutarne la robustezza. Infatti, una volta ideati e implementati il lavoro non è concluso. Gli algoritmi crittografici invecchiano, le prestazioni dei nostri computer migliorano e nuove tecniche di crittoanalisi vengono proposte in letteratura. Pertanto gli algoritmi crittografici devono essere continuamente testati, analizzati e manutenuti.

Un passo nella storia

La crittografia è una branca della matematica applicata che fonda le sue origini in un passato lontano. Come non ricordare il cifrario di Cesare usato da bambini per sostituire le lettere di un messaggio e renderlo apparentemente incomprensibile. Fino al 1900 le tecniche crittografiche utilizzate erano molto semplici e si basavano su sistemi monoalfabetici o polialfabetici a sostituzione, che potremmo definire varianti, più o meno complesse, del cifrario di Cesare.

Dal 1900 al 1950, si registrano invece importanti sviluppi in ambito crittografico, sviluppi spinti anche dall'avvento delle due guerre mondiali. Tra le innovazioni introdotte in questo periodo vale la pena di ricordare i primi cifrari basati su tecniche algebriche, quelli basati su macchine a rotori e quelli con chiavi di cifratura molto lunga e monouso.

Un consistente balzo in avanti lo si registra negli anni '70. Fino ad allora gli algoritmi crittografici noti erano catalogati come *simmetrici*, cioè algoritmi per i quali si cifra e si decifra sempre con la stessa chiave segreta – il lettore può immaginare questa operazione come il naturale gesto di aprire e chiudere la porta di casa, operazione eseguita sempre con la stessa chiave e senza la quale sarebbe impossibile entrare... a meno di forzare la porta. L'invenzione degli algoritmi crittografici *asimmetrici* ha portato interessanti novità. Questi ultimi, infatti, sfruttano tecniche di cifratura che fanno uso di una coppia di chiavi. Il concetto non è particolarmente intuitivo ai non addetti ai lavori. È un po' come se chiudessimo la porta di casa con una chiave e, per rientrare, dovessimo aprire la stessa con una seconda. Una chiude, l'altra apre. E le due chiavi sono differenti! Tutto ciò è reso possibile da semplici passaggi matematici che dimostrano elegantemente la validità e la forza di questi algoritmi.

I lavori pubblicati in letteratura in questo periodo hanno fatto la storia della crittografia. Alcuni sono stati utilizzati per decenni e pensionati negli anni 2000 dopo aver subito innumerevoli attacchi – e.g. Data Encryption Standard (DES) – altri sono ancora largamente utilizzati – e.g. RSA, e il protocollo di scambio di chiavi di Diffie–Hellman (DH). E per quale ragione nel 2020 dovremmo utilizzare algoritmi crittografici degli anni '70? Non ne esistono di più "nuovi"? Di più sicuri? Di più efficienti? Il crittografo, in questo caso, ragiona in modo differente. Più "vecchio" significa più studiato, più analizzato, maggiormente testato e probabilmente più sicuro. Quindi, se l'algoritmo non ha mostrato debolezze, perché cambiarlo?

Dagli anni 90 in poi, abbiamo assistito a una progressiva diffusione della crittografia. Fino ad allora essa era percepita come un argomento di nicchia, studiata e utilizzata da un ristretto numero di persone sia in ambito accademico, sia in ambito industriale. La progressiva diffusione di Internet nelle nostre case, anche in quelle dei non addetti ai lavori, ha fatto da trampolino di lancio alla crittografia e alle sue applicazioni. Vale la pensa di ricordare (a) la nascita dei protocolli di comunicazione sicura (Stallings 2017) – e.g. Secure Sockets Layer (SSL) e il suo successore Transport Layer Security (TLS) – cioè canali cifrati di comunicazione che consentono l'invio di dati sulla rete Internet in modo sicuro, canali che noi tutti utilizziamo per esempio quando comunichiamo con una banca online; (b) la nascita di nuovi algoritmi crittografici simmetrici – e.g. Advanced Encryption Standard (AES) – divenuti standard internazionali (Stallings 2017) e attualmente utilizzati per cifrare i dati presenti nei nostri smartphone o quelli immessi sulla rete Internet; (c) l'utilizzo di funzioni crittografiche per il controllo dell'integrità dei dati (Trappe et al. 2006) – le prime funzioni hash della famiglia Secure Hash Algorithm (SHA-0, SHA-1, SHA-2) e di quella Message-Digest Algorithm (MD2, MD4, MD5) – cioè algoritmi in grado di rilevare istantaneamente la modifica, volontaria o involontaria, di un solo bit all'interno di un file di dimensione arbitraria; (d) la diffusione degli algoritmi per la firma digitale (Trappe et al. 2006) – ad esempio il Digital

Signature Algorithm (DSA)[1] approvato dal NIST, oppure quelli che fanno uso di RSA, dei Merkle tree[2], delle Curve Ellittiche come ECDSA, etc.

La crittografia oggi

Oggigiorno gli smartphone, le app, il cloud, le crittomonete, i dispositivi dell'Internet delle cose (IoT), etc. sono i mezzi che hanno consentito una diffusione di massa della crittografia. Infatti, essa e utilizzata da tutti, spesso inconsapevolmente, più e più volte al giorno. La utilizziamo tutte le volte che accendiamo il nostro smartphone, che usiamo le app di messaggistica, che accediamo ai tornelli della metropolitana, che utilizziamo un dispositivo Bluetooth, che ci colleghiamo a una rete WiFi, che leggiamo un'e-mail, che facciamo un acquisto online, etc.

Insieme alla sua diffusione, sta crescendo anche la consapevolezza tra gli utenti dell'importanza della stessa. Questa importanza è nota da tempo in ambito scientifico, industriale e governativo. Navigando in rete, infatti, non è difficile trovare documenti a supporto di tale affermazione. Per esempio, un interessante documento della Central Intelligence Agency (CIA), intitolato "Encryption Policy" e datato Settembre 1996, sottolinea le aspettative di crescita della crittografia, le preoccupazioni relative alla difficoltà di decifrare messaggi cifrati e i possibili stratagemmi da adottare per mitigarne la forza. Le parole di John Deutche, allora direttore dell'agenzia, erano le seguenti[3]:

> "… The use of strong, affordable commercial encryption will grow, along with international electronic commerce. We want U.S. industry to retain its dominant market share by incorporating strong encryption into communications, computer systems, software products. Yet strong encryption can undermine law enforcement and national security by limiting U.S. and other governments' abilities to exploit communications intercepts and access computer files, consistent with the law. The policy we recommend charts a middle of the road approach that will promote encryption worldwide, but, we hope, limit the negative effects. The heart of the policy is to encourage U.S. industry to adopt an encryption recovery system. In this system, encryption keys are deposited with certified "trusted parties" either here or abroad, who would provide access to the key for authorized law enforcement purposes..."

A oltre vent'anni di distanza da questo documento, è facile osservare i profondi cambiamenti che erano stati ipotizzati. Siamo passati dagli acquisti effettuali in negozio ai portali dell'e-commerce online, dal noleggiare le videocassette dei nostri film preferiti agli abbonamenti a film/sport/documentari scaricabili via streaming, dagli SMS ai messaggi vocali con le nostre app preferite, dal denaro in contante alle crittomonete. Il balzo in avanti è stato considerevole e nuove sfide aspettano i crittografi.

Nuove sfide all' orizzonte

Dal 2017, la comunità crittografica sta affrontando due importanti sfide. In particolare, i nuovi contesti operativi – IoT e computer quantistici – stanno richiedendo uno sforzo significativo nella definizione di algoritmi crittografici con caratteristiche profondamente differenti rispetto a quelli studiati negli ultimi cinquant'anni. Per questa ragione la comunità crittografica sta proponendo, analizzando e testando gli algoritmi che verranno utilizzati da tutti i cittadini nelle decadi a venire, algoritmi che dovranno soddisfare nuovi contesti di utilizzo.

[1] https://csrc.nist.gov/Projects/digital-signatures
[2] https://patents.google.com/patent/US4309569
[3] https://web.archive.org/web/20121015182952/
http://www.foia.cia.gov/docs/DOC_0000239468/DOC_0000239468.pdf

 doi:10.4399/97888255301552

Proprio in questa direzione il National Institute of Standards and Technology (NIST) ha lanciato negli anni scorsi due gare non competitive relative al processo di standardizzazione della crittografia "lightweight"[4] e di quella "post-quantum"[5].

La prima gara, ha come obiettivo quella di fornire strumenti crittografici in grado di soddisfare le richieste dell'IoT. Infatti, le scarse risorse disponibili in questo ambito non consentono l'utilizzo degli attuali algoritmi crittografici che sono stati pensati per essere eseguiti su notebook, desktop, server e certamente non sono adatti all'utilizzo nei sensori delle smart home, nei componenti intelligenti delle nostre auto, nelle etichette RFID, etc. Gli algoritmi crittografici che vinceranno questa gara dovranno essere in grado di offrire un buon compromesso tra protezione delle informazioni e utilizzo delle (scarse) risorse messe a disposizione da tali dispositivi.

La seconda gara, va in direzione diametralmente opposta. Con l'avvento dei computer quantistici e delle loro capacità computazionali, quello che prima poteva essere considerato sicuro ora non lo è più. In particolare, tali computer consentiranno di attaccare i cifrari asimmetrici precedentemente menzionati. In questo caso, gli algoritmi sottoposti alla gara post-quantum dovranno essere robusti e non attaccabili neppure dai computer quantistici. Va sottolineato, invece, che gli attuali cifrari simmetrici soddisfano già questo requisito e che l'unica accortezza che dovremo adottare sarà quella di raddoppiare la dimensione delle chiavi di cifratura.

La robustezza degli algoritmi

Ma gli algoritmi crittografici che noi tutti utilizziamo sono sicuri? Per rispondere a questa domanda dobbiamo porcene una seconda. Stiamo parlando di sicurezza dell'algoritmo "sulla carta" oppure di una sua implementazione?

Partiamo dalla prima. Gli unici strumenti che abbiamo per misurare la forza di un algoritmo crittografico sono quelli matematici. Alcuni degli algoritmi che utilizziamo comunemente basano la loro forza sulla nostra incapacità di risolvere determinati problemi matematici in un tempo ragionevole: il problema della fattorizzazione di grandi numeri, il problema del logaritmo discreto, etc. Nessuno ha mai dimostrato che questi problemi sono impossibili da risolvere. Nonostante ciò, già dagli anni '70, i ricercatori hanno adottato tali problemi come punto di partenza per la progettazione di nuovi algoritmi crittografici. Una delle cose fondamentali da ricordare è che gli algoritmi crittografici, basati o non basati su problemi matematici difficili da risolvere, sono attaccabili a patto di avere una quantità illimitata di tempo o risorse. La tecnica utilizzerò per spiegare come eseguire un attacco non è tra le più sofisticate ma almeno non richiede conoscenze crittografiche pregresse. Infatti, all'attaccante basterà provare tutte le chiavi di cifratura o decifrazione finché non avrà trovato quella giusta… Ahimè, se dovessimo provarle tutte non otterremmo una risposta in tempo utile neppure utilizzando il più potente computer presente sulla faccia della Terra. Per quantificare questo sforzo basta giocare con le probabilità. Se prendessimo un messaggio cifrato con uno standard internazionale, ad esempio AES-256, che probabilità avremmo di decifrare tale messaggio provando pazientemente le chiavi per un periodo di tempo di 100 anni? Dopo qualche semplice conto, il lettore si accorgerà che tale probabilità è infinitesimamente piccola, talmente piccola da essere inferiore alla probabilità di vedere un grosso asteroide colpire la Terra nei prossimi due secondi… 1 secondo, 2 secondi… e come avrete notato, questo non è successo! E da cosa è data questa forza? Questa forza è data dal numero totale di operazioni di cifratura o decifrazione che dovremmo eseguire. Tale numero è molto grande. Anzi è astronomico! E' dello stesso ordine di grandezza del numero di particelle elementari presenti nell'universo. Ora non vi sarà difficile immaginare perché i crittoanalisti non adottano questa tecnica ma prendono altre vie...

Passiamo alla seconda. Adesso che abbiamo compreso perché gli algoritmi crittografici sono "sicuri" dal punto di vista teorico, cerchiamo di capire cosa succede alle loro implementazioni. Gli strumenti che abbiamo a disposizione in questo caso sono meno rigorosi. La cosa più semplice da fare è trovare un controesempio e cercando in rete se ne trovano a decine… Ahimè, gli errori di progettazione, di programmazione, di disattenzione, ma anche di scarse conoscenze sono all'ordine del giorno. Ovviamente questo non accade solo nelle piccole realtà ma anche a colossi internazionali. A titolo di esempio riportiamo due episodi curiosi. Il primo è quello capitato alla Microsoft mentre cercava di progettare una console per videogiochi (Xbox) che fosse utilizzabile

[4] https://csrc.nist.gov/projects/lightweight-cryptography
[5] https://csrc.nist.gov/Projects/post-quantum-cryptography/Post-Quantum-Cryptography-Standardization

solamente con sistema operativo e software originale[6,7]. In fase di progettazione, alla Microsoft decisero di utilizzare il cifrario TEA per eseguire gli opportuni controlli di integrità del software installato. Ahimè, TEA ha una nota debolezza, cioè quella di avere un sottoinsieme di chiavi equivalenti e pertanto non deve essere usato come funzione hash per eseguire controlli di integrità (Kelsey et al. 1996). Se così non fosse, un attaccante potrebbe sostituire il software originale con uno scelto a piacere, facendo credere all'algoritmo adibito al controllo d'integrità che nulla è stato modificato... Il secondo episodio è quello capitato alla Sony e, per pura coincidenza, anche qui parliamo di una console per videogiochi (PlayStation)[8]. In questo caso l'errore commesso è legato al cattivo utilizzo dei parametri crittografici utilizzati per firmare digitalmente il software originale. L'algoritmo crittografico utilizzato richiede l'uso di un numero casuale, non predicibile, che deve essere cambiato ad ogni utilizzo. Dopo la prima generazione, tale numero non è stato cambiato, consentendo quindi agli attaccanti di recuperare la chiave privata dalla Sony attraverso semplici passaggi algebrici. Una volta scoperta, questa chiave consentiva di firmare digitalmente qualsiasi software (incluso quello non originale) impersonificando la Sony...

E se la chiave è corta?

Un ragionamento differente lo si deve fare quando cerchiamo di misurare la forza di cifrari aventi chiavi relativamente corte. Infatti, in questo caso è possibile montare un attacco di forza bruta basato sulla ricerca esaustiva delle chiavi. Un famoso esempio è quello del cifrario DES (Trappe et al. 2006). Nel 1977, pochi mesi dopo la sua standardizzazione, la comunità crittografica aveva ipotizzato che specifici computer costruiti ad-hoc erano in grado di attaccare DES nel giro di 24 ore. Questi computer, dal costo stimato di 20 milioni di dollari, non erano certo alla portata dei comuni cittadini, ma sicuramente lo potevano essere per le agenzie governative. Negli anni successivi DES entrò in una di quelle fasi di revisione periodica e programmata ai quali i cifrari sono sottoposti. In particolare, nel 1987 la discussione se ricertificare DES come standard entrò nel vivo. Vennero sottolineati i primi segni di debolezza del cifrario dati dall'accresciuta forza computazionale dei computer dell'epoca e contemporaneamente venne proposto di rimpiazzare DES con un insieme di algoritmi progettati della National Security Agency (NSA). Questa proposta fu bocciata e DES venne ricertificato. Nel 1992, durante il successivo periodo di revisione, nulla cambiò, ma nel 1996 tre nuove modalità di attacco ai cifrari simmetrici vennero proposte in letteratura. La prima si basava su hardware dedicato – questo approccio aveva il vantaggio di essere particolarmente performante ma lo svantaggio di essere tremendamente costoso. La seconda modalità di attacco faceva uso della forza computazionale dei computer di molti utenti, suggerendo quindi la realizzazione di un attacco distribuito – questo approccio era il meno costoso di tutti, ma aveva lo svantaggio di non essere particolarmente performante dal momento che i computer utilizzati erano di tipo "general purpose". La terza ed ultima proposta era un'implementazione ibrida delle due precedenti. In tutti e tre i casi l'obiettivo era quello di eseguire un attacco di forza bruta sull'intero insieme delle chiavi.

Nel 1997, la RSA Data Security lanciò una sfida. Avrebbe pagato 10.000 dollari alla prima persona che sarebbe riuscita ad attaccare un messaggio cifrato con DES. Grazie alla crescente diffusione di Internet, il secondo approccio prese il sopravvento. In particolare, Rocke Verser scrisse un programma che sfruttava i computer della rete volontariamente ceduti in prestito dai loro proprietari. E per quale ragione gli utenti avrebbero dovuto prestare a Verser le proprie macchine? Per raccogliere un largo consenso, egli promise che il 60% della vincita sarebbe stata sua, mentre il restante 40% sarebbe finito nelle tasche di quell'utente che, mettendo a disposizione la propria macchina, avrebbe trovato la chiave. Venne così implementato il primo attacco distribuito e tale attacco consentì la scoperta della chiave corretta dopo aver provato circa il 25% delle chiavi possibili. Erano già passati 5 mesi dal lancio della sfida, ma una precisa strategia si era fatta strada: l'attacco distribuito.

L'anno seguente, la RSA Data Security lanciò una seconda sfida identica alla precedente. Questa volta bastarono solo 39 giorni per rompere il messaggio cifrato analizzando circa l'85% delle chiavi. Infine, nel 1998, l'Electronic Frontier Foundation con un investimento di 200.000 dollari costruì un hardware aspecifico e riuscì a dimostrare che era in grado di attaccare messaggi cifrati con DES in meno di 5 giorni. Nello stesso anno, il NIST bandì una nuova gara crittografica

[6] https://events.ccc.de/congress/2005/fahrplan/attachments/591-paper_xbox.pdf
[7] https://www.schneier.com/blog/archives/2005/08/x-box_security.html
[8] https://www.bbc.com/news/technology-12116051

 doi:10.4399/97888255301552

per individuare il sostituto di DES e il vincitore venne chiamato Advanced Encryption Standard (AES).

Debolezze nascoste?

Il punto di forza dei processi di standardizzazione fatti dal NIST è quello di mettere sotto la lente di ingrandimento le nuove proposte provenienti dalla comunità scientifica. In queste gare, gli autori spiegano il razionale che sta alla base di determinate scelte, forniscono un'implementazione di riferimento liberamente scaricabile e suggeriscono una serie di possibili ottimizzazioni della stessa. Tutti possono analizzare i dettagli dell'algoritmo. Tutti possono controllarne la sicurezza. Chiunque può sollevare delle critiche. In questo modo, la comunità scientifica vuole evitare gli spiacevoli effetti collaterali dati dalla presenza di backdoor nascoste in algoritmi proprietari o mai resi completamente pubblici. Questo però non vuol dire che all'interno di algoritmi noti, e largamente studiati, non si possono inserire delle backdoor (Young et al. 1997). A titolo di esempio, riportiamo un famoso caso che ha visto coinvolta la NSA. La problematica che andremo a illustrare è legata alle Curve Ellittiche (EC) e in particolare al Dual Elliptic Curve Deterministic Random Bit Generator (Dual ECDRBG). Già nel giugno del 1999[9] era sorta la preoccupazione che tale curva, divenuta successivamente standard ANSI, ISO e NIST, fosse in qualche modo "cotta a puntino" per facilitare l'inserimento di una backdoor. Infatti, il processo di generazione della curva in questione adottava un particolare numero casuale il cui razionale era rimasto inspiegato, offrendo così la possibilità di generare curve ellittiche manipolabili[10,11]. Nel 2013, un articolo pubblicato da "The Guardian"[12] riferiva che tra documenti trapelati da Edward Snowden c'era evidenza del fatto che l'NSA aveva lavorato per inserire una backdoor nello standard Dual ECDRBG[13]. Questa backdoor avrebbe consentito all'NSA di decifrare i canali cifrati di comunicazione sicura (SSL/TLS) che utilizzavano tale curva come generatore di numeri casuali. Nell'aprile 2014, il NIST rimosse tale curva dal suo paniere suggerendo di passare a uno degli altri algoritmi approvati[14] ma l'accaduto mostrò all'intera comunità scientifica quanto fosse concreta la possibilità di nascondere una trapdoor all'interno di un cifrario.

Bibliografia

Kelsey, John, Bruce Schneier, and David Wagner. "Key-schedule cryptanalysis of idea, g-des, gost, safer, and triple-des." *Annual International Cryptology Conference*. Springer, Berlin, Heidelberg, 1996.

Stallings, William. *Cryptography and network security: principles and practice*. Upper Saddle River: Pearson, 2017.

Trappe, Wade, and Lawrence C. Washington. *Introduction to cryptography with coding theory*. Pearson Education, 2006.

Young, Adam, and Moti Yung. "Kleptography: Using cryptography against cryptography." In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Berlin, Heidelberg, 1997.

[9] https://groups.google.com/forum/message/raw?msg=sci.crypt/mFMukSsORmI/FpbHDQ6hM_MJ
[10] http://safecurves.cr.yp.to/rigid.html
[11] https://miracl.com/blog/backdoors-in-nist-elliptic-curves/
[12] https://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security
[13] https://bits.blogs.nytimes.com/2013/09/10/government-announces-steps-to-restore-confidence-on-encryption-standards/
[14] https://www.nist.gov/news-events/news/2014/04/nist-removes-cryptography-algorithm-random-number-generator-recommendations