Un'analisi delle tecniche di crittografia "white box"

Gregory R. Ellis
Vice president operation
Product line manager
Microsemi PMG Security Solutions

All'aumentare delle minacce e degli attacchi strettamente correlati alla diffusione della tecnologia IoT (Internet of Things), la crittografia "white box" deve essere considerata un elemento essenziale per proteggere le operazioni crittografiche in qualsiasi sistema software

In questo articolo sono analizzate diverse tecniche di crittografia di tipo "white box", utilizzate per proteggere dati e operazioni crittografiche all'interno di ambienti dove è possibile che si verifichino attacchi di tipo "white box". La necessità dell'uso della crittografia "white box", la descrizione delle tecniche e delle metodologie di una tipica implementazione della crittografia "white box", l'esame delle modalità utilizzate dalla crittografia "white box" per prevenire attacchi contro dati e operazioni crittografiche critiche e le principali caratteristiche di una realizzazione "white box" sono gli argomenti trattati nel corso dell'articolo.
Vista la crescente esigenza di adottare la crittografia software e il contemporaneo aumento di minacce e attacchi strettamente correlati alla diffusione della tecnologia IoT (Internet of Things), la crittografia "white box" deve essere considerata un elemento essenziale per proteggere le operazioni crittografiche in qualsiasi sistema software.

La necessità della crittografia "white box"

I numeri della crescita di Internet of Things sono sicuramente da record. Con una stima di oltre 200 miliardi di unità connesse entro il 2020, non c'è dubbio che i dispositivi collegati a Internet stiano influenzando quasi tutti gli aspetti della nostra vita. Internet of Things è una tecnologia che incide su una molteplicità di mercati, dalla robotica ai sistemi PoS (Point of Sales), dai dispositivi di elaborazione mobili alla stampa 3D. I sistemi embedded destinati a questi mercati svolgono una pluralità di funzioni: tengono informati gli utilizzatori, prendono decisioni autonome, comunicano con i partner commerciali e sono persino in grado di gestire le finanze personali.
L'accesso ai dati, ai sistemi informativi e ai contenuti digitali presenti su questi sistemi sono di solito protetti mediante cifratura. Per proteggere le informazioni cifrate è indispensabile che la chiave crittografica utilizzata per la cifratura di tali dati non venga mai rivelata. Nelle implementazioni di crittografia tradizionale sia la chiave sia l'algoritmo sono vulnerabili a fenomeni di manomissione e di "reverse engineering" – per ogni sistema crittografico il singolo punto di vulnerabilità – SPOF (Single Point of Failure) si verifica quando la chiave viene utilizzata. Questo punto di vulnerabilità è facilmente identificabile nei moderni sistemi utilizzando analisi di firma, di pattern e della memoria. Ad esempio, attacchi per l'estrazione della chiave, condotti contro chiavi codificate sotto forma di serie di dati letterali in software non protetti, possono essere condotti e completati con esito positivo nel giro di poche ore.

Uno sguardo sulla crittografia "white box"

La crittografia "white box" è un metodo molto ben documentato, utilizzato per offuscare un algoritmo crittografico in modo che il materiale della chiave sia sufficientemente al riparo da occhi indiscreti. Obbiettivo della crittografia "white box" è impedire che le informazioni critiche (come ad esempio la chiave) relative alle operazioni di crittografia siano rivelate a un potenziale aggressore che ha accesso completo al sistema.
Il nome crittografia "white box" trae la sua origine da un particolare tipo di attacco denominato appunto attacco "white box". Contrariamente a quanto accade nel caso di attacchi "black box", nei quali un aggressore non ha accesso al sistema, negli attacchi "white-box" il potenziale aggressore ha il completo accesso al sistema, alla sua memoria, alle sue routine software e così via. Si può affermare che, nel momento in cui i sistemi sono sempre più aperti e mobili (laptop, tablet, cellulari), diventano accessibili con maggior facilità e risultano quindi più vulnerabili agli attacchi "white box" se non si prendono misure di sicurezza adeguate.
Un algoritmo "white box" è solitamente offuscato in modo tale che l'accesso o la conoscenza dell'implementazione

non possano compromettere il materiale della chiave anche durante le operazioni crittografiche. Una tipica implementazione "white box" di uno standard di crittografia prevede la cifratura, decifrazione, firma e verifica dei dati sensibili, come accade in una realizzazione di tipo tradizionale, ma costringere l'aggressore a eseguire il reverse engineering di trasformazioni matematiche complesse per ottenere la chiave segreta (secret key).
L'implementazione "white box" risulta pertanto utile laddove la crittografia deve essere effettuata in un ambiente potenzialmente soggetto a vulnerabilità dove è necessario proteggere le chiavi crittografiche e/o i dati in chiaro (plaintext) oppure nei casi in cui un potenziale utente "non affidabile" potrebbe assumere il controllo del sistema host. In scenari di questo tipo si potrebbero verificare problemi di notevole entità quali ad esempio la compromissione di sistemi connessi in rete, la possibilità che il software venga reso disponibile alla concorrenza o l'installazione di software commerciale con chiavi private (con la conseguente possibilità di poter eseguire su una macchina qualsiasi tipo di codice).

Prevenzione degli attacchi con la crittografia "white box"

Un esempio di attacco di vasta portata è quello che nell'aprile 2014 ha sfruttato la vulnerabilità Heartbleed presente in OpenSSL, uno dei software di crittografia più popolari e diffusi (basti solamente pensare che esso è utilizzato nel web server Apache e, quindi, su circa due terzi dei server web di tutto il mondo, per la gestione delle connessioni SSL/TLS attraverso il protocollo HTTPS). Un attacco che sfrutta questo bug consente la lettura di porzioni di memoria del server che potrebbero contenere parte del materiale della chiave crittografica utilizzata per rendere sicure le comunicazioni tra quel server e il mondo esterno. L'esposizione della chiave rischia di compromettere i dati (molto sensibili) protetti da quel canale di comunicazione.
Per proteggere le informazioni cifrate è indispensabile che la chiave non si manifesti in alcun modo nella memoria o sull'hard disk. Nelle implementazioni crittografiche standard sia l'algoritmo sia la chiave sono vulnerabili a manomissioni e operazioni di "reverse engineering". La crittografia "white box" trasforma per via matematica la chiave in un complesso grafo di numeri e codice eseguibile. Nel grafo in questione esistono più percorsi validi scelti in maniera casuale durante l'esecuzione sulla base di una sorgente di numeri random fornita dall'utente (Fig. 1).
L'abbinamento tra algoritmi matematici, dati e tecniche di oscuramento del codice per trasformare la chiave e le relative operazioni di crittografia seguendo modalità complesse richiede una profonda conoscenza in molteplici discipline da parte dell'aggressore. Un aspetto particolarmente importante è il fatto che la chiave non è mai presente nella memoria statica o durante l'esecuzione (run-time). La chiave è semplicemente una raccolta di dati che non hanno nessuna utilità se non si dispone dell'algoritmo "white box" generato "ad hoc". In breve, sostituendo le librerie crittografiche standard con una libreria in grado di supportare la tecnica "white-box" le chiavi non saranno mai esposte, annullando di fatto l'efficacia di attacchi del tipo descritto.

```
White-Box AES Key:
0000000 0e18 80bc bae7 e250 708d ea28 04dd 9a18
0000010 f615 0d93 cf64 b9b6 7c5c 73be 2282 2176
0000020 d870 ade7 656b c188 48a4 cbe0 6ec6 9f1f
0000030 2c54 dd21 30f3 bdc7 d438 3b61 6850 8094
0000040 83e7 d907 57e8 db00 a39a 1ddb 59ec 7e29
0000050 6bae fd3d b2b0 604e edf7 98a3 c519 56c4
0000060 deb8 93c4 432d 4146 9fa6 5637 9d8e d7df
0000070 986e b925 d5a4 81ed c4c6 3778 9cc8 aa8b
0000080 8006 3b73 2a7e 87d9 3248 157c b5f6 f9b3
0000090 c30c 3a62 0dbe 5bb4 0cc7 f788 664e 2f69
00000a0 57d0 ad7d 0b70 1a92 b251 efb3 60c0 bdf4
00000b0 27d2 ebdf 916d dae5 c981 be66 667c c9cc
00000c0 2634 e17c 082d d0f8 338f 3e58 c9ee 3780
00000d0 2a01 9224 6d71 6344 66bb b037 5e96 2320
00000e0 13d7 d7aa 9f42 f210 5dfa 66b0 dc5b 070e
00000f0 a2dc 5fb3 7e53 bd5e 0830 e021 83cf 3764
0000100 e870 30a5 3320 8d0b aa3b f86a 3a75 e71c
0000110 5e85 84e8 1db4 6d82 0ee4 c64a 1bf7 2657

Based on the Classical 256-bit AES Key:
0000000 e502 d48a 18d7 95cd 5992 b8b0 d88b 65f1
0000010 78e8 264f 3652 bb4b fbb9 6802 c914 c4d0
```

Fig. 1 – La relazione tra la chiave classica e quella dell'implementazione "white box" non è banale, rendendo di fatto impraticabile la ricostruzione della chiave classica utilizzando i tool solitamente disponibili per un aggressore che conduce attacchi di tipo "netwok based" (ovvero che sfruttano il traffico sulla rete)

Tecniche da utilizzare in un'implementazione White Box

Pur nella loro diversità, le tecniche di seguito descritte sono di fondamentale importanza per ciascuna implementazione di tipo white-box da utilizzare in sistemi potenzialmente soggetti a vulnerabilità.

Diversità

Invece di implementare un algoritmo di crittografia "white box" per tutti gli utilizzatori (che potrebbe essere vulnerabile ad attacchi di tipo Bore – Break-once-run-everywhere - copia una volta, esegui dappertutto), i generatori di codice dovrebbero essere utilizzati per produrre varianti uniche dell'algoritmo. Ciò permette di semplificare l'indagine preliminare di dati sensibili (come ad esempio chiavi o testo in chiaro selezionato). Algoritmi realizzati "ad hoc" consentono

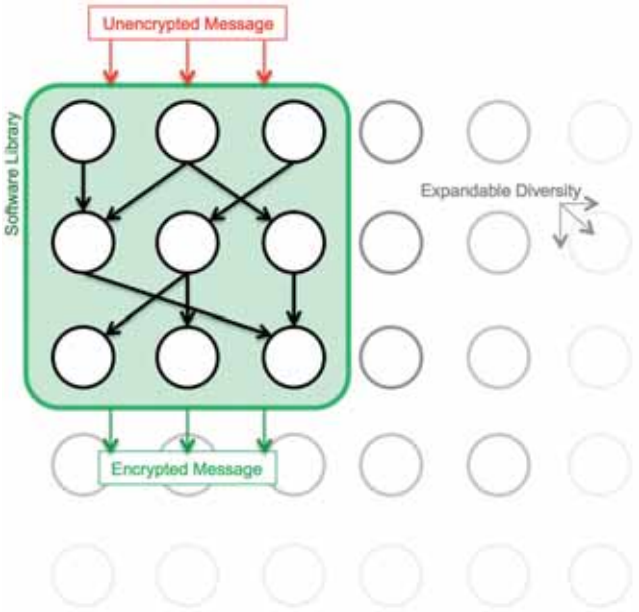

Fig. 2 – Un'implementazione "white box" dovrebbe prevedere la randomizzazione e l'offuscamento di più trasformazioni che sfruttano i principi matematici alla base dell'algoritmo di crittografia

anche di eliminare attacchi algebrici che potrebbero riuscire a "svelare" le protezioni dei dati grazie alla comprensione di una singola implementazione standard.
Gli algoritmi potrebbero essere implementati utilizzando metodi matematici alternativi (Fig. 2). Gli algoritmi "white box" non dovrebbero essere trasformazioni automatiche di algoritmi standard. Ciascun algoritmo/cifrario dovrebbe essere modificato in maniera tale da sfruttare le specifiche proprietà dei principi matematici alla base dell'algoritmo stesso: una trasformazione in blocco non dovrebbe mai essere applicata su tutti gli algoritmi.

Binding dell'hardware

Per sua natura, il software è più vulnerabile agli attacchi rispetto all'hardware. Mediante la semplice copiatura, bit per bit, del software originale, un aggressore avrebbe a disposizione un numero virtualmente illimitato di tentativi per aprire una breccia nel software. Un'implementazione efficace e robusta della crittografia "white box" dovrebbe, quando possibile, sfruttare l'hardware per limitare la possibilità che vengano eseguite operazioni di reverse engineering sugli algoritmi offuscati.
Una di queste tecniche prevede il bindind dell'hardware. Dal punto di vista della crittografia, il binding (ovvero l'associazione) di un identificatore dell'hardware a un algoritmo "white box" e/o ai dati costringe un aggressore ad effettuare il reverse engineering di un grafo della chiave molto complesso che varia in maniera dinamica collegato a un singolo sistema hardware.

Resistenza agli attacchi side-channel

La resistenza contro gli attacchi side-channel (ovvero basati su informazioni collaterali) come ad esempio quelli di tipo SPA (Simple Power Analysis) o DPA (Differential Power Analysis) che sfruttano l'analisi delle tracce dei consumi di potenza è di fondamentale importanza per proteggere dall'esposizione il materiale della chiave. Un'implementazione affidabile della crittografia "white box" dovrebbe utilizzare numerose contromisure contro analisi di tipo "side-channel" per evitare l'esposizione della chiave nel corso di questi attacchi.

Metodi di offuscamento

Alcuni attacchi contro molti algoritmi crittografici potrebbero produrre risposte note. Molte volte, il progetto di algoritmi di crittografia standard ha dato come risultato implementazioni vulnerabili agli attacchi di tipo "white box" in quanto basati sull'implicita ipotesi che questi algoritmi vengano eseguiti su host sicuri. Un'implementazione "white box" robusta dovrebbe eliminare vulnerabilità di questo tipo. L'offuscamento "white box" dovrebbe attenuare il rischio che attacchi lanciati da aggressori esperti nei principi matematici su cui si fonda l'algoritmo possano convincere con l'inganno gli algoritmi a produrre una versione offuscata di una risposta ben nota. Inoltre, l'uso di tecniche di offuscamento come quella di evanescenza dei confine di un ciclo (round boundary blurring) permette di nascondere punti di attacco ben definiti che potrebbero compromettere un ciclo di elaborazione (round) dell'algoritmo AES.
In definitiva, grazie alla crittografia "white box" le chiavi non sono disponibili per eventuali aggressori che sono quindi costretti a effettuare operazioni di "reverse engineering" di numerose e complesse combinazioni di trasformazioni di offuscamento: gli aggressori, inoltre, devono possedere solide conoscenze nel settore dell'algebra astratta e della matematica discreta. In considerazione dell'aumento esponenziale dei dispositivi mobili connessi a Internet e alla sempre maggiore necessità di poter effettuare operazioni e comunicare in modo sicuro, un'implementazione della crittografia "white box" utilizzando le tecniche descritte nel corso dell'articolo deve essere vista alla stregua di un elemento critico di ogni sistema software che utilizza la crittografia. ■

Bibliografia

1. *Wyseur, Brecht. "White-Box Cryptography: Hiding Keys in Software." MISC Magazine. April 2012.*
2. *Saxena, Wyseur, Preneel. "Towards Security Notions for White-Box Cryptography."*
3. *Kocher, Jaffe, Jun. "Introduction to Differential Power Analysis and Related Attacks." 1998.*