Protezione del Software

Alfredo De Santis
Dipartimento di Informatica
Università di Salerno

ads@dia.unisa.it
http://www.dia.unisa.it/professori/ads

Marzo 2012

Protezione dalla copia

- Trovare un metodo contro la pirateria
  - efficiente
  - economico
  - resistente contro i pirati esperti
  - non invasivo
- Compito impossibile!
- Però si può rendere la copia difficile per il pirata esperto ed impossibile per il pirata occasionale

Indice

- Numeri di serie
- Implementazione dei numeri di serie
- Dongle
- Esecuzione remota
- Difesa dalle contraffazioni
  - Checksum
  - Esche
  - Offuscamento

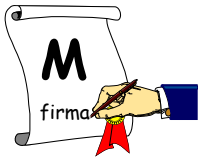
Numeri di serie

- Il software necessita di un numero di serie per funzionare

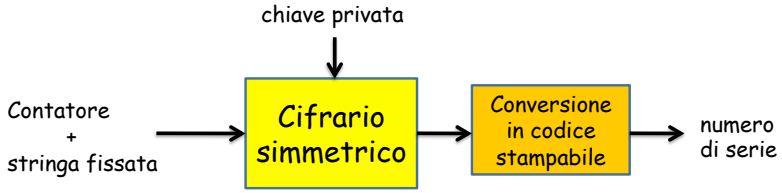
- Molto utilizzato dalle aziende di software
- Non offre un alto grado di protezione
- Ma è un deterrente psicologico
  - Digitare un numero di serie *copiato* rende consapevoli di ciò che si sta facendo

Implementazione dei numeri di serie

- Cifrari simmetrici
- Firme digitali



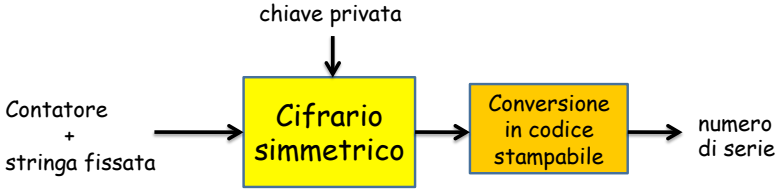
Implementazione dei numeri di serie con cifrari simmetrici



Nel software vi sono:

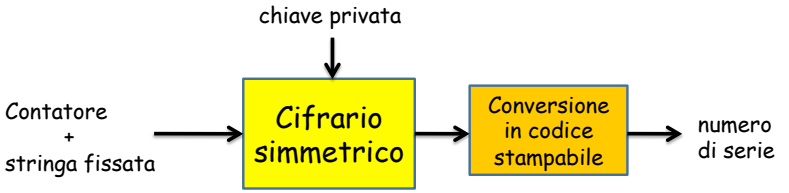
- chiave privata
- stringa fissata
- implementazione inversi di "Cifrario" e "Conversione"

Implementazione dei numeri di serie con cifrari simmetrici

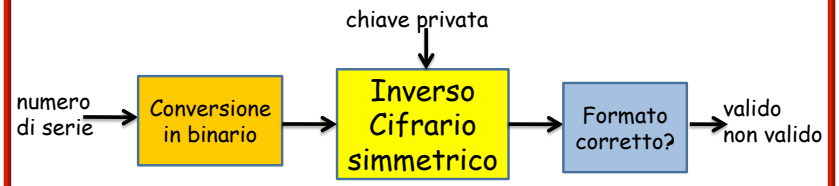


Come avviene il controllo di validità?

Controllo validità



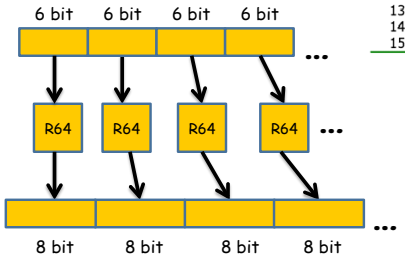
Controllo validità del numero di serie



Conversione codice stampabile

- Base64
- Set di caratteri formato ASCII RADIX-64

valore	codifica	valore	codifica	valore	codifica	valore	codifica
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/



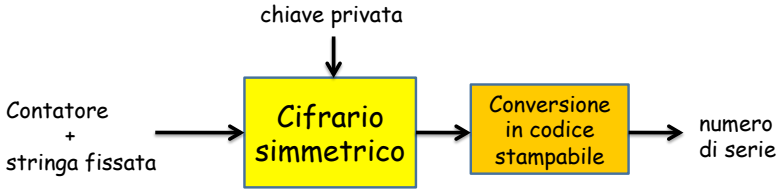
Espansione 33%

Conversione codice stampabile

- Base64
  - Facile conversione
  - Scomodo per alcuni utenti (causerebbe errori battitura)
- Per evitare errori di battitura, è meglio
  - Ignorare distinzione maiuscole/minuscole
- Uso di alfabeto romano e 10 cifre (totale 36 caratteri)
- Però è utile avere alfabeto potenza di 2
- Eliminare 4 caratteri, ad es. l 1 o 0 (facili da confondere)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	B	C	D	E	F	G	H	I	J	K	M	N	P	Q	R	S	T	U	V	W	X	Y	Z	2	3	4	5	6	7	8	9

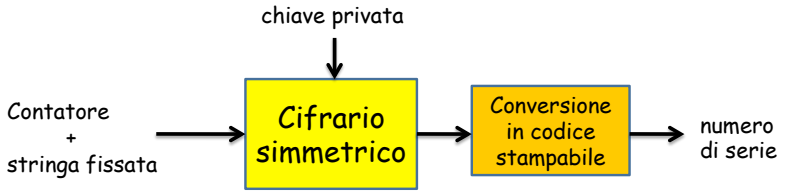
Stringa fissata



Identifica il programma, ed anche la versione

Chiavi di versioni precedenti non devono funzionare anche per le nuove versioni

Implementazione dei numeri di serie con cifrari simmetrici: Sicurezza



Quanto è sicuro?

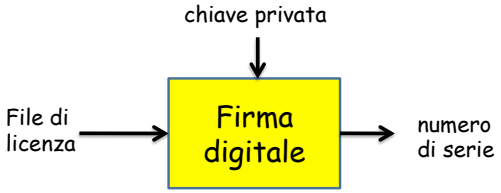
Implementazione dei numeri di serie con cifrari simmetrici: Sicurezza

- Possibile "trovare" *chiave privata* e *stringa fissata* nel codice binario
- Possibile "generare" numeri di serie validi

- Scopo dei numeri di serie:
  Scoraggiare solo i pirati occasionali

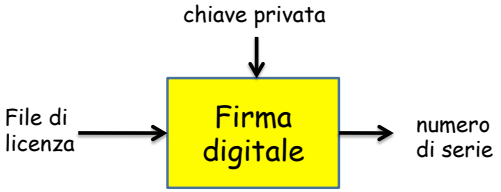
- Non tiene lontani i pirati esperti

Implementazione dei numeri di serie con firme digitali



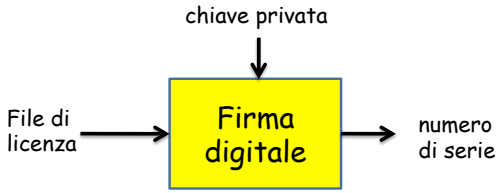
- Nel software vi sono:
  - chiave pubblica
  - implementazione verifica "Firma digitale"
- Non c'è la chiave privata!

Implementazione dei numeri di serie con firme digitali

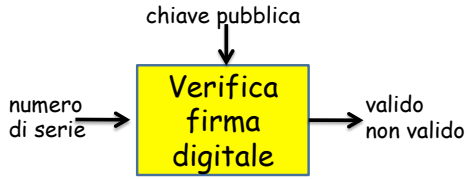


Come avviene il controllo di validità?

Controllo validità



Controllo validità del numero di serie



File di licenza

- Informazioni sulle licenze:
  - Informazioni specifiche dell'utente (ad es., nome)
  - Versione del software con alcune caratteristiche a pagamento

Inserimento dati licenze

- Numeri di serie sono chiamati anche *chiavi di licenze*
- Inserimento di chiavi di licenze e file di licenza
  - Digitate dall'utente
  - File posizionato dove il software può trovarlo

Implementazione dei numeri di serie con firme digitali: Sicurezza

- Non ci sono chiavi private nel codice binario
- Non è possibile "generare" numeri di serie validi

- Possibile inserire un numero di serie che si conosce
- Attacchi di decompilazione

Licenze con validità limitata nel tempo

- Software shareware
- Aggiungere data nel file di licenza
- Per verificare se la licenza è scaduta:
  Confronto con data del sistema
- Metodo classico per aggirare il controllo:
  Cambiare la data del sistema

Licenze con validità limitata nel tempo

- Software shareware
- Aggiungere data nel file di licenza
- Per verificare se la licenza è scaduta:
  Confronto con data del sistema
- Metodo classico per aggirare il controllo:
  Cambiare la data del sistema
- Contromisura: Rendere persistenti i dati

  Ad es., mettere in un registro: data prima esecuzione, data ultima esecuzione, MAC delle due date

Licenze con validità limitata nel tempo

- Eliminazione dati oppure
  Disinstallazione e poi Installazione
- Contromisura: Si possono nascondere i dati nel file system in modo che non possono essere cancellati
- Se si cancellassero *tutti* i dati alla disinstallazione un attaccante potrebbe scoprire dove si nascondono
- Un programma che lascia tracce è una sciagura per gli utenti legittimi
- Bisogna pensare bene alle conseguenze di tali schemi, prima di adottarli!

Elevare il livello di protezione

- Per usare illegalmente il software su più macchine: copiare il numero di serie
- Contromisura: includere dati specifici della macchina nella licenza

  Per es., quantità di memoria, indirizzo scheda Ethernet
- Tecnica intrusiva

  Il proprietario non può modificare troppo la configurazione della propria macchina

Elevare il livello di protezione

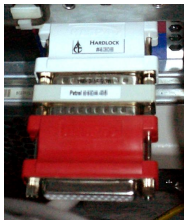
- Per applicazioni Internet:
  Callback a un server centrale, che ha traccia delle licenze utilizzate
- Problemi:
  - Privacy dell'utente
  - Si deve disporre di una connessione ad Internet (ad es., difficile su un aereo)
  - Se si dovesse reistallare il software
    (Si può permettere la reistallazione e poi monitorare comportamenti sospetti)
  - Firewall

Un vecchio metodo di protezione dalla copia

- Porre domande che richiedono di consultare la documentazione per rispondere

  Ad es., Qual è la terza parola, rigo 12, pag. 489, vol. 4?
- Un attaccante avrebbe dovuto copiare tutta la documentazione cartacea
- Oggi è tutto in formato elettronico, facile da copiare
- Anche se fosse tutto in formato cartaceo, basterebbe acquisirli una volta con uno scanner e diffonderli su Internet

Dongle



- Chiamati anche: hardware key, hardware token, o security device
- Si collegano alla porta seriale o parallela (poi USB) e sono dotati di limitate capacità di elaborazione
- Applicazione verifica presenza dongle durante l'esecuzione, inviando richieste e ricevendo risposte. Ad esempio:
  - Computazione valore funzioni
  - Esecuzione di parte del codice
- Compito attaccante: riprodurre in forma software il codice del dongle

Dongle

- Importante stabilire quale codice inserire nel dongle
- Ci sono limitazioni di capacità di elaborazione e memoria
- Non è impossibile capire cosa fa il dongle per un attaccante esperto
- Problemi:
  - Soluzione costosa
  - Scomoda per l'utente (detestati da chi li ha usati)

Esecuzione remota

- Molto simile ai dongle
- Ogni client
  - ha delle credenziali
  - si autentica su un server remoto
  - esegue parte del codice sul server remoto
- Vantaggi: più economico dei dongle
- Svantaggi:
  - più scomodo, perché necessita di un componente online
  - Produttore software deve gestire le credenziali e mantere server ad alta disponibilità

Difesa dalle contraffazioni

- L'attaccante potrebbe analizzare l'eseguibile e modificarlo
- Contromisura: Rendere gli eseguibili difficili da analizzare e da modificare

Checksum

- Calcolare checksum di dati e routine che possono essere attaccati
  - E' il valore precalcolato uguale al valore calcolato dinamicamente?
- Un attaccante potrebbe capire che il codice protetto dal checksum è importante
- Contromisura: più checksum, anche su parti secondarie, con controlli incrociati e dipendenti

Esche

- Aggiungere esche con controlli di licenza validi e facili da trovare
- Un attaccante è indotto a pensare che la protezione sia più debole di quella reale
- Introdurre checksum su questo codice esca
- Se l'esca viene rimossa o modificata, introdurre sottili bug nel programma

Offuscamento

- Aggiungere codice
  - Codice che non viene mai eseguito
  - Codice che non fa nulla
  - Rendere i calcoli più complessi del necessario
  - Utilizzare relazioni matematiche per condizioni vero/falso (attaccante non sa che è sempre vera/falsa)
- Spostare il codice
  - Copiare e rinominare la stessa funzione, invee di richiamarla da più punti
  - Unire più funzioni in una sola, assicurandosi la corretta esecuzione dei blocchi di codice in base alla chiamata
- Codificare i dati in modo strano
  - Conversioni in set di caratteri strano
  - Creazione di stringhe stampabili solo se necessario
  - Cifrare i dati
- Avvertenze
  - Rallentamento esecuzione
  - Programmazione di bassa qualità
  - Valutare bene le conseguenze!

Cifratura

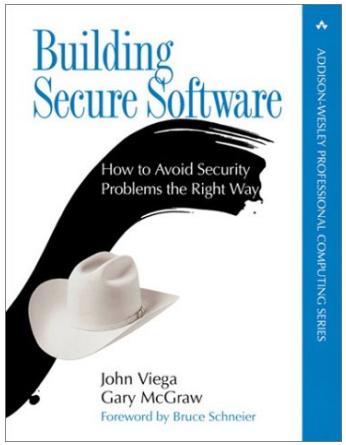
- Cifrare parte del codice e dei dati
- Il codice ha la procedura di decifratura con la chiave privata

Conclusioni

- Trovare un metodo contro la pirateria
  - efficiente
  - economico
  - resistente contro i pirati esperti
  - non invasivo

  è un compito impossibile!
- Però si può cercare di rendere lo sforzo richiesto per la contraffazione più grande del vantaggio corrispondente
- Abbiamo descritto alcune tecniche
- Si possono applicare molte diverse tecniche in modo giudizioso!

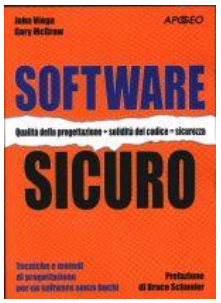
Bibliografia



www.buildingsecuresoftware.com

Si trova anche il codice del testo

Capitolo 15



Domande?