IN450 Algoritmi per la Crittografia

A.A. 2018/2019

Prof. Marco Pedicini

1. Crittografia Classica

- Crittosistemi di base: cifratura per sostituzione, per traslazione, per permutazione, affine, di Vigenère, di Hill. Cifratura a flusso (sincrona e asincrona), Linear feedback shift registers (LFSR) su campi finiti, Cifrario autokey. Cifrari prodotto. Crittoanalisi di base: classificazione degli attacchi; crittoanalisi per i cifrari affini, per la cifratura a sostituzione (analisi delle frequenze), per la cifratura di Vigenère: Kasiski test, indice di coincidenza; crittoanalisi del cifrario di Hill e degli LFSR: attacchi algebrici, cube attack.

2. Applicazione della Teoria di Shannon alla crittografia

- Sicurezza dei cifrari: sicurezza computazionale, sicurezza dimostrabile, sicurezza incondizionata. Richiami di calcolo delle probabilità: variabili aleatorie discrete, probabilità congiunta, probabilità condizionata, variabili aleatorie indipendenti, Teorema di Bayes. Variabili aleatorie associate a crittosistemi. Sistemi di cifratura a sicurezza perfetta. Crittosistema di Vernam. Entropia. Codici di Huffman. Spurious Keys e Unicity distance.

3. Cifrari a blocchi

- Schemi di cifratura iterativi; Reti di Sostituzione-Permutazione (SPN); Crittoanalisi lineare per SPN: Piling-Up Lemma, approssimazione lineare di S-boxes, attacchi lineari a S-boxes; Crittoanalisi differenziale per SPN; Cifrari di tipo Feistel; DES: descrizione e analisi; AES: descrizione; Cenni sui campi finiti: operazioni su campi finiti, algoritmo di Euclide generalizzato per il calcolo del mcd e degli inversi; Modi operativi per i cifrari a blocchi.

4. Funzioni Hash e Codici per l'autenticazione di messaggi

- Funzioni di hash e integrità dei dati. Funzioni di hash sicure: resistenza alla controimmagine, resistenza alla seconda controimmagine, resistenza alla collisione. Il modello dell'oracolo random: funzioni di hash ideali, proprietà di indipendenza. Algoritmi randomizzati, collisione sul problema della seconda controimmagine, collisione sul problema della controimmagine. Funzioni di hash iterate; la costruzione di Merkle-Damgøard. Algoritmo di Hash Sicuro (SHA-1). Codici di Autenticazione (MAC): codici di autenticazione nidificati (HMAC).

Testi consigliati

[1] Joux, A., *Algorithms for Cryptanalysis*. CRC Press, (2010).
[2] Stinson, D. R., *Cryptography: Theory and Practice*. Chapman & Hall/CRC, (2002).
[3] Delfs H., Knebl H., *Introduction to Cryptography*. Springer-Verlag, (2007).

Bibliografia supplementare

[4] Lidl, R. and Niederreiter, H., *Finite Fields*. Cambridge University Press, (2007).
[5] Mangano, S., *Mathematica Cookbook*. O'Reilly, (2014).
[6] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, *Handbook of Applied Cryptography*. CRC press, (1997).
[7] Song Y. Yan, *Number Theory for Computing*. Springer, (2002).
[8] Neal Koblitz, *Algebraic Aspects of Cryptography*. Springer, (1998).
[9] Neal Koblitz, *A Course in Number Theory and Cryptography* . Springer, (1994).
[10] Bruce Schneier, *Applied Cryptography*. Wiley, (1996).
[11] Niels Ferguson, Bruce Schneier, *Practical Cryptography*. Wiley, (2003).
[12] Ross Anderson, *Security Engineering*. Wiley, (2001).

Modalità d'esame

- valutazione in itinere ("esoneri")		■ SI	□ NO
- esame finale	scritto	■ SI	□ NO
	orale	□ SI	■ NO
- altre prove di valutazione del profitto (meglio descritte sotto)		■ SI	□ NO

L'esame consiste di due parti: un esame scritto e un progetto di programmazione.

Gli argomenti del progetto di programmazione devono essere concordati con il docente e consistono nella implementazione delle funzioni di cifratura, e di decifratura del crittosistema e nella implementazione di uno degli attacchi noti. Può essere eseguito in un linguaggio a scelta dello studente tra Java, C, C++, Mathematica. Il progetto può essere presentato prima o dopo il superamento dello scritto.

La prova orale è prevista per riparare le insufficienze lievi allo scritto.