Criptografie și securitate CTI

Laborator 11

Funcții hash. Message Authentication Codes (MACs). Cifrare autentificată

1. Răspundeți cu adevărat sau fals pentru fiecare dintre următoarele afirmații. Căutați online informații despre funcțiile hash menționate.
   (a) Amestecarea ingredientelor pentru realizarea unei prăjituri poate fi considerată one-way function.
   (b) Funcția hash MD5 este considerată sigură la coliziuni.
   (c) SHA256 este o funcție hash cu output pe 256 biți.
   (d) Valoarea hash SHA-1 pentru cuvantul "laborator" este $0x4bcc6eab9c4ecb9d12dcb0595e2aa5fbc27231f3$.
   (e) Este corect să afirmăm că "o funcție hash criptează".
   (f) O funcție hash folosită pentru stocarea parolelor trebuie să fie rapidă (i.e., să se calculeze rapid H(x) pentru x dat).
   (g) Hash-ul (fără salt) - $095b2626c9b6bad0eb89019ea6091bd9$ – corespunde unei parole sigure, care nu ar fi susceptibilă spre exemplu la un atac de tip dicționar.
2. (a) Ce este un algoritm de tip MAC? Ce asigură acest tip de primitivă criptografică? Dați minim două scenarii în care pot fi utilizați algoritmii de tip MAC.
   (b) Arătați că CBC-MAC nu este o construcție sigură dacă este utilizată pentru autentificarea mesajelor de lungimi diferite.
3. (a) Ce asigură un algoritm criptografic din categoria celor de cifrare autentificată?
   (b) Descrieți cele 3 moduri de criptare autentificată de la link-ul următor `https://en.wikipedia.org/wiki/Authenticated_encryption`.
4. Arătați cum se poate realiza semnătura digitală a algoritmului RSA folosind funcții hash. Verificați dacă insecuritățiile variantei de semnături digitale identificate în laboratorul anterior se regăsesc și în această schemă.
5. Citiți despre familia de funcții hash PHOTON. În continuare, problema se referă la varianta cu output pe 80 biți. Urmăriți pașii de mai jos:
   (a) Generați 2 fișiere (input.txt) de 1 000 000, respectiv 10 000 000, de linii diferite două câte două;
   (b) Descărcați Reference-Implementation.zip de pe Teams și integrați acest cod într-un proiect propriu;

(c) Calculați valoarea hash a fiecărei linii create în fișierele anterioare și verificați dacă există vreo coliziune cu vreuna dintre valorile hash din fișierul output_test.txt. În caz afirmativ returnați valoarea pentru care se întâmplă coliziunea.

Link suplimentar: `https://sites.google.com/site/photonhashfunction/design`