Ataki kryptograficzne.

Krótka historia kryptografii...

- Szyfr Cezara
  A -> C
  B -> D
  C -> E
  ...
  X -> Z
  Y -> A
  Z -> B

- ROT13 - pochodna szyfru Cezara nadal używana
  ROT13(ROT13("Tekst jawny") = "Tekst jawny".

- permutacje - *26! kluczy* = 403.291.461.126.605.635.584.000.000.
- ...
- *DES - Data Encryption Standard* - klucz 56 bitowy
- *AES - Advanced Encryption Standard* - wybrany w 2001r. jako następca *DES‘a*.
- *RSA*
- szyfrowanie kwantowe.

Co to jest kryptoanaliza??

Kryptoanaliza - zdobycie jak najpełniejszej informacji o tekście jawnym bez znajmości tajnego klucza.

Zasady:

- Przeciwnik zawsze zna wykorzystaną metodę.

- Bez znajomości kryptoanalizy opracowanie dobrego algorytmu szyfrującego nie ma żadnego praktycznego sensu.

- Nikt nie jest w stanie samodzielnie przetestować algorytmu w dostatecznym stopniu. Algorytm musi być zaprezentowany publicznie, aby na całym świecie można go było przedyskutować.

Przykłady ataków kryptograficznych:

- Atak przy użyciu tekstu tajnego (ciphertext only attack).
  Klucz lub tekst jawny zdobywane są wyłącznie przy wykorzystaniu krypto-gramu. Jest to najtrudniejsza metoda.

- Atak przy użyciu tekstu jawnego (known-plain-text-only attack).
  Oprócz kryptogramu znany jest pewien fragment tekstu jawnego. Za jego pomocą wydobywa się resztę treści, z reguły poprzez znalezienie klucza.

- Atak przy użyciu wybranego tekstu jawnego (chosen-plaintext attack).
  Jest to również atak przy użyciu tekstu jawnego, jednak atakujący ma moż-liwość podsunięcia fragmentu tekstu jawnego.

- Atak przy użyciu dopasowanego wybranego tekstu jawnego (adaptive-chosen-plaintext attack).
  Jest to wielokrotnie powtarzany atak przy użyciu wybranego tekstu jawnego. W tym przypadku każdy następny tekst jawny podsuwany do zaszyfrowania jest wybierany w zależności od dotychczasowych rezultatów kryptoanalizy.

Sposoby łamania.

- *frequency analysis* - metoda wykorzystująca częstość występowania liter.
- *ciphertext relative length analysis* - metoda wykorzystująca dlugość zakodowanego tekstu.
- *similar plaintext analysis* - metoda badająca zaszyfrowane podobne informacje.

Poszukiwany algorytm szyfrujący:

- *konfuzja* - wykrycie zależności między tekstem jawnym a kryptogramem powinno być niemożliwe. (oczywiste)
- *dyfuzja* - struktury tekstu jawnego powinny zostać w jak najwyższym stopniu zamazane.
- *długość klucza* powinna być wielka, by atak typu *brute force* wiązał się ze zbyt dużymi nakładami w stosunku do wartości wiadomości.

- Z takich samych bądź podobnych tekstów jawnych nie powinne nigdy zostać wygenerowane takie same lub podobne kryptogramy.

- Kryptogram nie może statystycznie odróżniać się od ciągu liczb losowych.

- w wypadku stałych, cyklicznych lub w inny sposób charakterystycznych tekstów jawnych nie powinny występować dostrzegalne cykle.

- *efekt lawinowy* w wyniku zmiany dowolnego bitu w tekście jawnym każdy bit kryptogramu powinien zmienić swój stan z prawdopodobieństwem równym dokładnie 50%.

- ataki przy użyciu znanego tekstu jawnego nie powinny dawać żadnych praktycznych rezultatów.

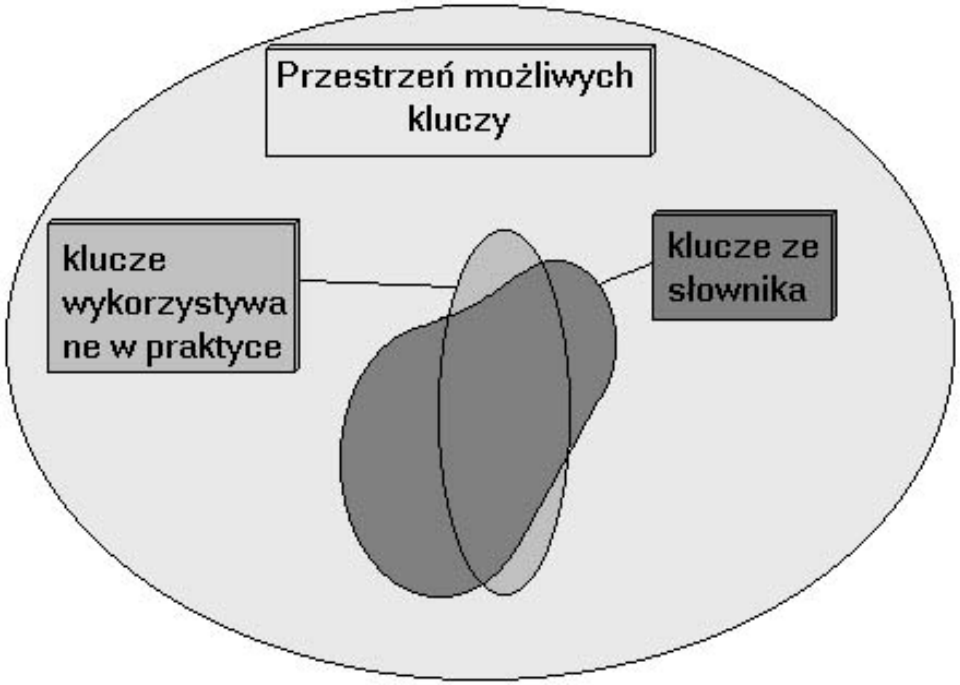
- ... i inne ...

Brute force:

- zalety:
  Zawsze działa. (np. został użyty do złamania DES-a)

- wady:
  Przeważnie działa długo.

Atak słownikowy.



Wybrane programy:

- *L0phtcrack* - Windows‘ach NT.
  Przy użyciu 4 x Xeon 400
  - hasła alfa-numeryczne - 5.5 godziny
  - hasła alfa-numeryczne-dodatkowe symbole - 45 godzin
  - hasła alfa-numeryczne-wszystkie symbole - 480 godzin
- LC3 - dodano rozproszone łamanie haseł

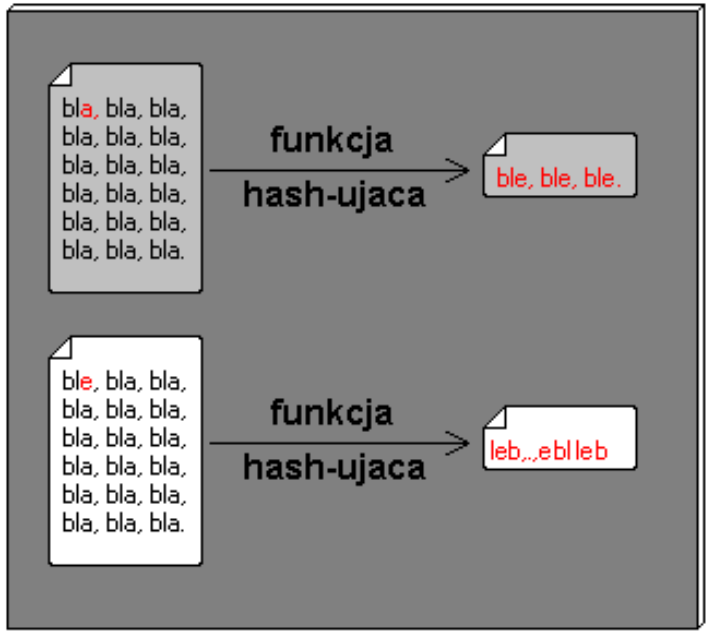
- *Crack* - najstarszy i najczęściej używany program do łamania haseł pod UNIX-em

- *John The Ripper* - dostępny pod DOS‘em, UNIX‘em i Windows‘em. Używany do łamania haseł szyfrowanych przez MD5.

Funkcje *hash*‘ujące wykorzystywane w szyfrowaniu.

- jednostronna
- łatwo obliczalna
- nieciągła
- małe prawdopodobieństwo kolizji

Funkcje *hash*‘ujące wykorzystywane w szyfrowaniu c.d.



Robaki - *Worms* - historia.

- Brunner - 1975 - *tapeworm* - pierwszy robak program niszczący sieć komputerową, skierowany przeciw agencji rządowej, napisany przez John‘a Brunnera.

- Morris - 1988 - *RTM* - robak wykorzystujący dziury w *sendmail‘u i fingerd*, zarażał setki maszyn, autor Robert Tappan Morris student Cornell University został skazany na 400 godzin prac publicznych, 10.000$ i 3 lata nadzoru.

- ...

- ADM - 1998 - *ADMwOrm* - tworzy on kopię *shell'a* a także nowe konto, logi zostają skasowane autor dostaje informacje o zdobyciu kolejnej maszyny, *robak* ten zapoczątkował podmienianie plików *index.html*, opublikowane zostają źródła co wywołuje ogromne zainteresowanie.

- *Melissa & I Love You* - oszacowane straty 8.000.000$, dobry pomysł na rozprzestrzenianie się.

- *Code Red Worm i inne...*