Institute of Operating Systems
and Computer Networks

Technische
Universität
Braunschweig

Security with noisy Data

Organisation

Stephan Sigg

March 30, 2011

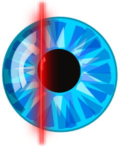
Outline

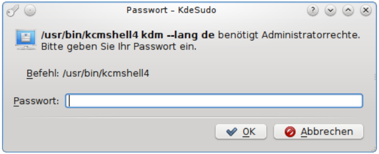
Overview and Structure

- Classification methods
- Feature extraction
  - Features from audio
  - Features from RF
- Fuzzy Commitment
- Fuzzy Extractors
- Authentication with noisy data
- Error correcting codes
- Entropy
- Physically unclonable functions

Introduction

- Nowadays, security is an important integral part of many applications and services
  - Passwords to authenticate at devices and services
  - Encryption mechanisms to secure communication links, speech and data
- In order to increase security measures and to improve usability, biometric information is also utilised
  - Fingerprints
  - Iris scan





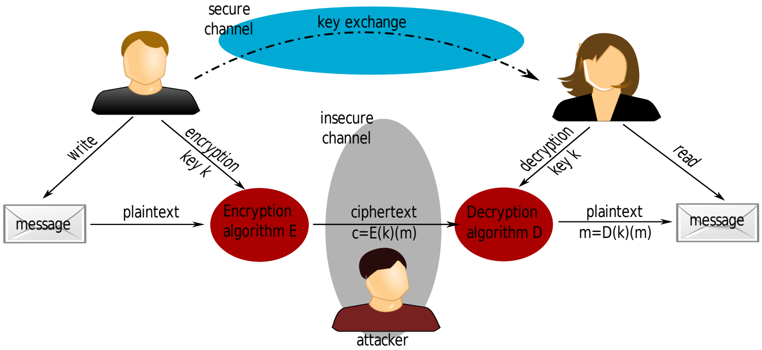
Introduction

- We distinguish between
  - Symmetric encryption techniques
  - Asymmetric encryption techniques

Symmetric encryption

- Symmetric encryption is a very old concept.
- All participants in the communication have to know the common secret
- A symmetric encryption function is applied
  - The encryption function is also utilised for decryption
- Not well suited for distributed systems that communicate over an insecure channel
- Well suited for password creation
  - Unlikely that the secret is stolen during password creation

Symmetric encryption

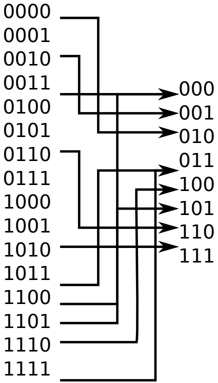


Symmetric encryption

- Typically, the password is not stored as plain text
- Often, the hash of the password is stored
- A one-way hash function is utilised
  - The hash function introduces a second layer of security
  - An attacker that would gain access to the hash stored by the system can still not obtain the password easily

Symmetric encryption

- A hash function maps a large amount of data into a small datum
- Usually a single integer
- May serve as an index to an array



Symmetric encryption

Example

Unix or Linux systems

- The hash of the password is usually stored to `/etc/passwd`
- When a user picks a password, it is encoded with a randomly generated value called the salt
- This means that any particular password could be stored in 4096 different ways
- The salt value is then stored with the encoded password

Symmetric encryption

Example

Unix or Linux systems

- When a user logs in and supplies a password, the salt is first retrieved from the stored encoded password.
- Then the supplied password is encoded with the salt value,
- and then compared with the encoded password.
- If there is a match, then the user is authenticated.

Symmetric encryption

Example

Unix or Linux systems

- Computationally difficult to take a randomly encoded password and recover the original password.
- On a system with multiple users, at least some passwords are common words (or simple variations of common words).
  - Encrypt a dictionary of words and common passwords using all possible 4096 salt values and compare the encoded passwords in `/etc/passwd` file with this database (dictionary attack).
  - One of the most common methods for gaining or expanding unauthorized access to a system.

Symmetric encryption

Example

Unix or Linux systems

- `/etc/passwd` contains user and group ID's for many programs
  - Therefore, it must remain world readable.

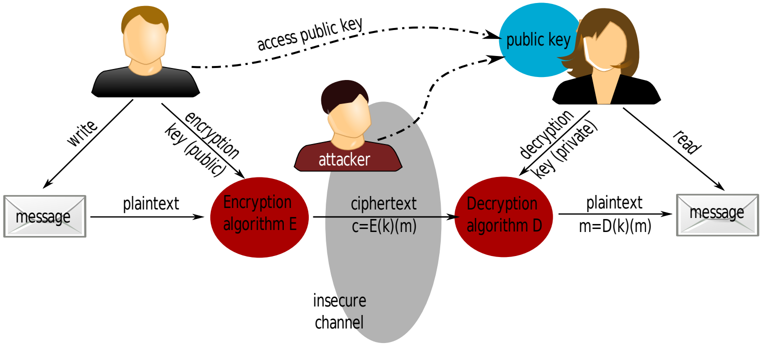
Symmetric encryption

Example

Unix or Linux systems

- This problem can be solved by using the shadow suite
  - Relocates passwords to another file (e.g. `/etc/shadow`)
  - `/etc/shadow` cannot be read by just anyone
  - Only root has read/write access to `/etc/shadow`
- Programs that only need to verify passwords can either be run suid root or a group shadow can be utilied that is allowed read only access to the `/etc/shadow`.

Asymmetric encryption



Asymmetric encryption

Example

RSA key generation

1. Select two large random prime numbers p and q
2. Compute $n = p \cdot q$
3. Compute $\theta(n) = (p-1) \cdot (q-1)$
4. Select small odd integer e that is relatively prime to $\theta(n)$

Asymmetric encryption

Example

RSA key generation

5. Compute d as the multiplicative inverse of $e \mod \theta(n)$
6. Publish $P = (e, n)$ as the RSA public key
7. Keep the secret pair $S = (d, n)$ as the RSA secret key

Asymmetric encryption

Example

RSA key generation

8. Encrypt a message M
   - $C = M^e \mod n$
9. Decrypt a message C
   - $M = C^d \mod n$

Asymmetric encryption

RSA key generation – example

- Select two prime numbers p and q
  - $p = 3; q = 11$
- Compute $n = p \cdot q$
  - $n = 33$
- Compute $\theta(n) = (p-1) \cdot (q-1)$
  - $\theta(n) = 20$

Asymmetric encryption

RSA key generation – example

$p = 3; q = 11; n = 33; \theta(n) = 20$

- Select a small odd integer e that is relatively prime to $\theta(n)$
  - $e = 3$
- Compute d as the multiplicative inverse of $e \mod \theta(n)$
  - $d = 7$
  - Test: $3 \cdot 7 \mod 20 = 21 \mod 20 = 1$

Asymmetric encryption

RSA key generation – example

$p = 3; q = 11; n = 33; \theta(n) = 20; e = 3; d = 7$

- Publish $P = (e, n)$ as the public key
  - Key pair: $(3, 33)$
- Keep $S = (d, n)$ as the RSA secret key
  - Secret key pair: $(7, 33)$

Asymmetric encryption

RSA key generation – example

$p = 3; q = 11; n = 33; \theta(n) = 20; e = 3; d = 7$

$P = (3, 33)$

$S = (7, 33)$

- Encrypt the message '3':
  - $C = M^e \mod n$
  - $C = 3^3 \mod 33 = 27 \mod 33 = 27$

Asymmetric encryption

RSA key generation – example

$p = 3; q = 11; n = 33; \theta(n) = 20; e = 3; d = 7$

$P = (3, 33)$

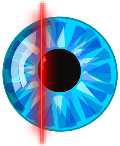
$S = (7, 33)$

- Decrypt the message $C = 27$:
  - $M = C^d \mod n$
  -

$$\begin{aligned} M &= 27^7 \mod 33 \\ &= 10460353203 \mod 33 \\ &= 3 \end{aligned}$$

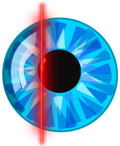
Introduction

- For the use of biometric information, we can not use the same concepts for the maintenance of the secret
  - It turns out that for biometric information nowadays the original sample is often stored on the device
  - However, concepts for security with noisy data have been developed



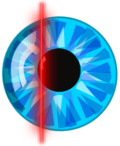
Using biometric features for authentication

- Increasingly, biometric features are utilised for authentication
  - Fingerprint
  - Iris scan
  -
- In a similar manner, electronic devices can be identified
  - e.g. with small fluctuations in their current consumption
  - or with electromagnetic radiation pattern



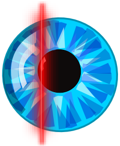
Using biometric features for authentication

- When these patterns are utilised for authentication or encryption, the fingerprint has to be stored somehow on the device
- Typically, authentication mechanisms with such features have to be tolerant for noise end errors in the feature
  - Errors in the scanned fingerprint
  - Inaccurate scanning hardware
  - Differing environmental circumstances



Using biometric features for authentication

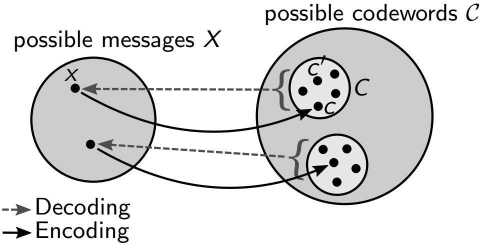
- Therefore, it is not possible to only store the hash value of the originally sampled feature
- When the feature is noisy, the hash function will produce a differing hash value at each application
- It is therefore required that the original feature is somehow stored on the system



Using biometric features for authentication

Fuzzy cryptography

- It is possible to utilise error correcting codes to account for errors in an input sequence
- The general idea is to utilise a function that maps from a feature space to another, smaller key space



Introduction

- We find noisy data in many typical application fields
  - Wireless sensor networks
  - Mobile communication
  - PUFs
  - Biometric data

Overview and Structure

- Classification methods
- Feature extraction
  - Features from audio
  - Features from RF
- Fuzzy Commitment
- Fuzzy Extractors
- Authentication with noisy data
- Error correcting codes
- Entropy
- Physically unclonable functions

Outline

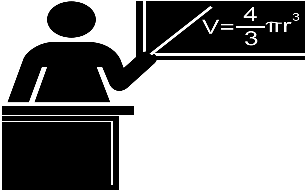
Introduction

Lectures and Excercises

Conclusion

Objectives

- Acquire detailed knowledge on methods for secure communication based on noisy input data
  - General principle
  - Algorithms and implementation
  - Various input data sources
- Practical experience of the lecture topics in hands-on projects



Requirements and lecture material

Requirements to successfully complete the lecture :

- Interest
- Ability to work self-employed but in teams
- Ask !!! when you do not understand something
  - In the lecture
  - In the exercise
  - Via Email

Material :

- http://www.ibr.cs.tu-bs.de/courses/ss11/noisy/
  - Lecture slides
  - Additional information

Organisation

Lecture : Tuesday, 13:15 - 14:45

Exercises : Tuesday, 15:00 - 16:30

- Every two weeks
- First exercise on April, 12th, 2011

Oral examination : 29.08.2011 - 02.09.2011

Outline

Questions?

Stephan Sigg
sigg@ibr.cs.tu-bs.de

Literature

- P. Tulys, B. Skoric, T. Kevenaar, Security with Noisy Data – On private biometrics, secure key storage and anti-counterfeiting, Springer, 2007.