Selected Topics of Pervasive Computing

Stephan Sigg

Georg-August-University Goettingen, Computer Networks

15.01.2014

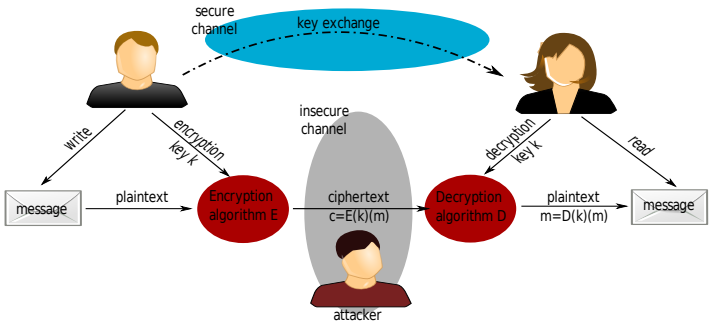
Overview and Structure

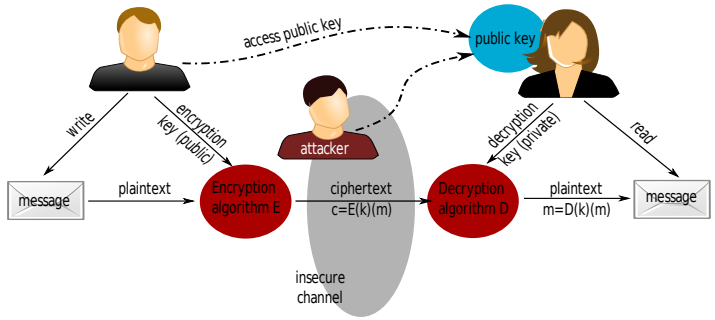
30.10.2013	Organisational
30.10.3013	Introduction
06.11.2013	Classification methods (Feature extraction, Metrics, machine learning)
13.11.2013	Classification methods (Basic recognition, Bayesian, Non-parametric)
20.11.2013	–
27.11.2013	–
04.12.2013	–
11.12.2013	Classification methods (Linear discriminant, Neural networks)
18.12.2013	Classification methods (Sequential, Stochastic)
08.01.2014	Features from the RF channel (Effects of the mobile radio channel)
15.01.2014	Security from noisy data (Encryption schemes, Fuzzy extractors)
22.01.2014	Security from noisy data (Error correcting codes, PUFs, Applications)
29.01.2014	Context prediction (Algorithms, Applications)
05.02.2014	Internet of Things (Sensors and Technology, vision and risks)

Outline

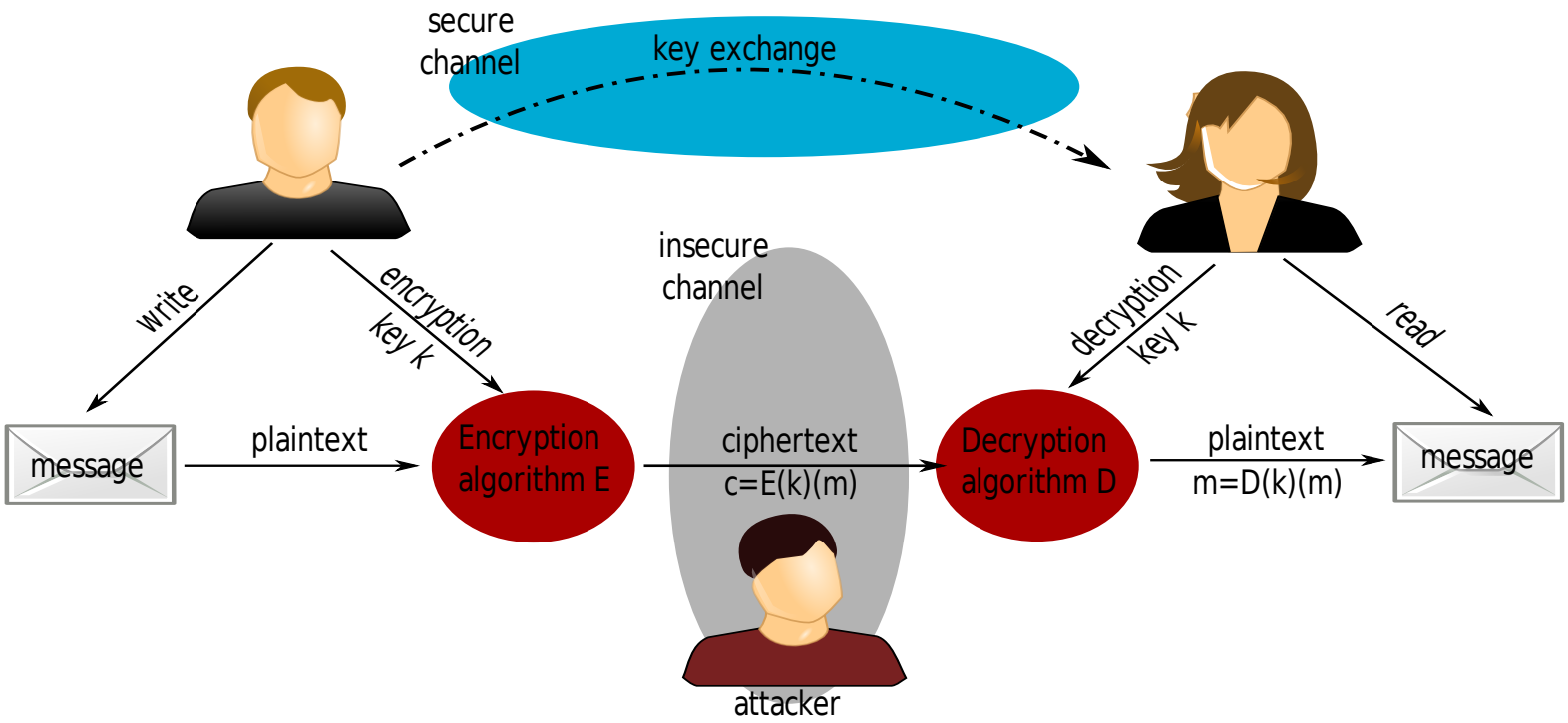
Introduction

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





Symmetric encryption

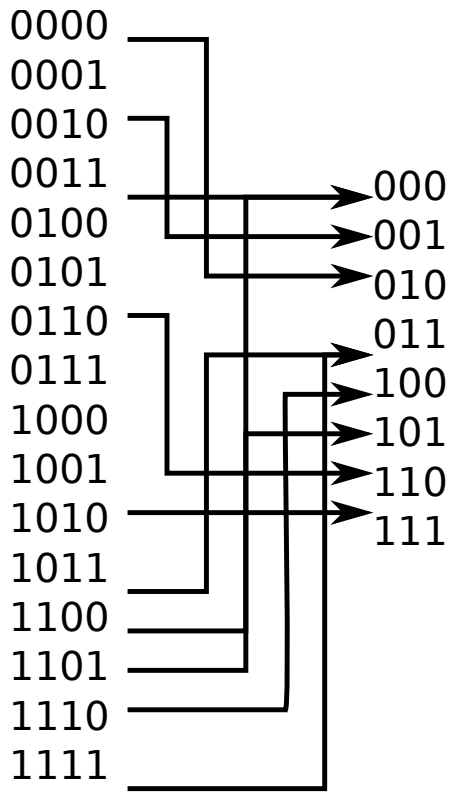


Symmetric encryption

- 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

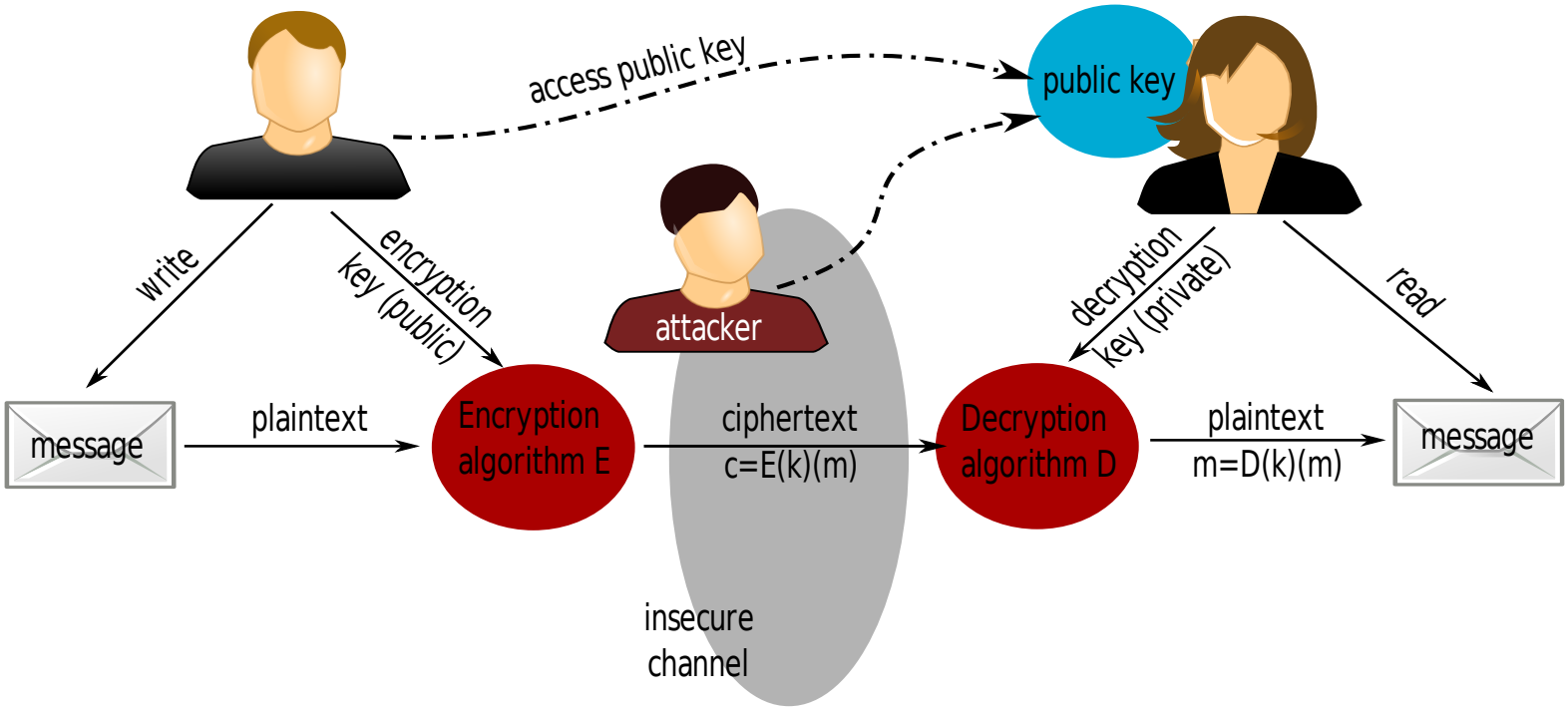
- A hash function maps a large amount of data into a small datum
- Usually a single integer
- May serve as an index to an array



Symmetric encryption

- For Symmetric encryption all participants in the communication have to know a common secret
- A symmetric encryption function is applied
  - The encryption function is also utilised for decryption
- Well suited for local password creation
  - Unlikely that the secret is stolen during password creation
- Not well suited for distributed systems that communicate over an insecure channel
  - How should two devices initially agree on a common secret?

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

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

$P = (3, 33)$

$S = (7, 33)$

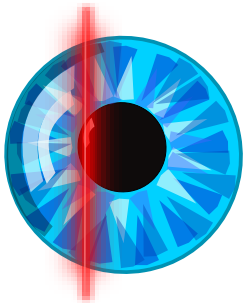
- Decrypt the message $C = 27$:
  - $M = C^d \mod n$
  -

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

Encryption and decryption in the presence of noise

With the increasing number of sensors and biometric information being integrated in security schemes, the key or the seed to the key is implicitly noisy

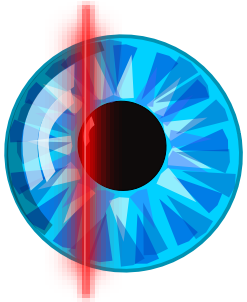
- We find noisy data in many typical application fields
  - Wireless sensor networks
  - Mobile communication
  - PUFs
  - Biometric data



Encryption and decryption in the presence of noise

When encountering noise in the input data, we can not use the same concepts for the maintenance of a secret

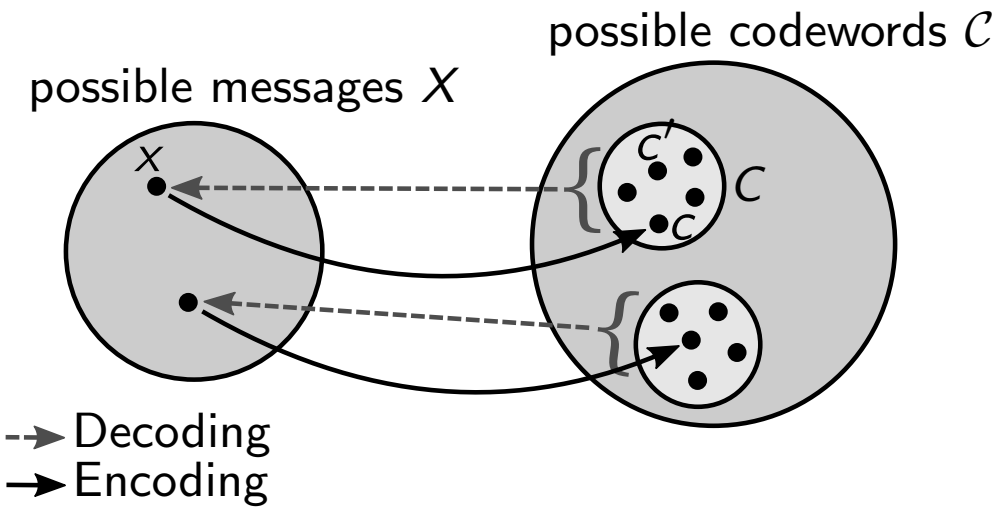
- It is not possible to only store the hash value of the originally sampled feature
- When the feature is noisy, the hash function will produce a differing hash value at each application
- It is therefore required that the original feature is somehow stored on the system



Encryption and decryption in the presence of noise

Fuzzy cryptography

- We can, however, 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, key space



Outline

Features of the RF channel

Features specific for the RF-channel

- Wlan Access points
- Signal Strength
- Signal to noise ratio
- Fluctuation in signal strength
- Energy on several frequency bands
- Active Bluetooth devices
- GSM base stations/GSM active set
- ...

Features of the RF channel

Secure communication based on RF-channel information

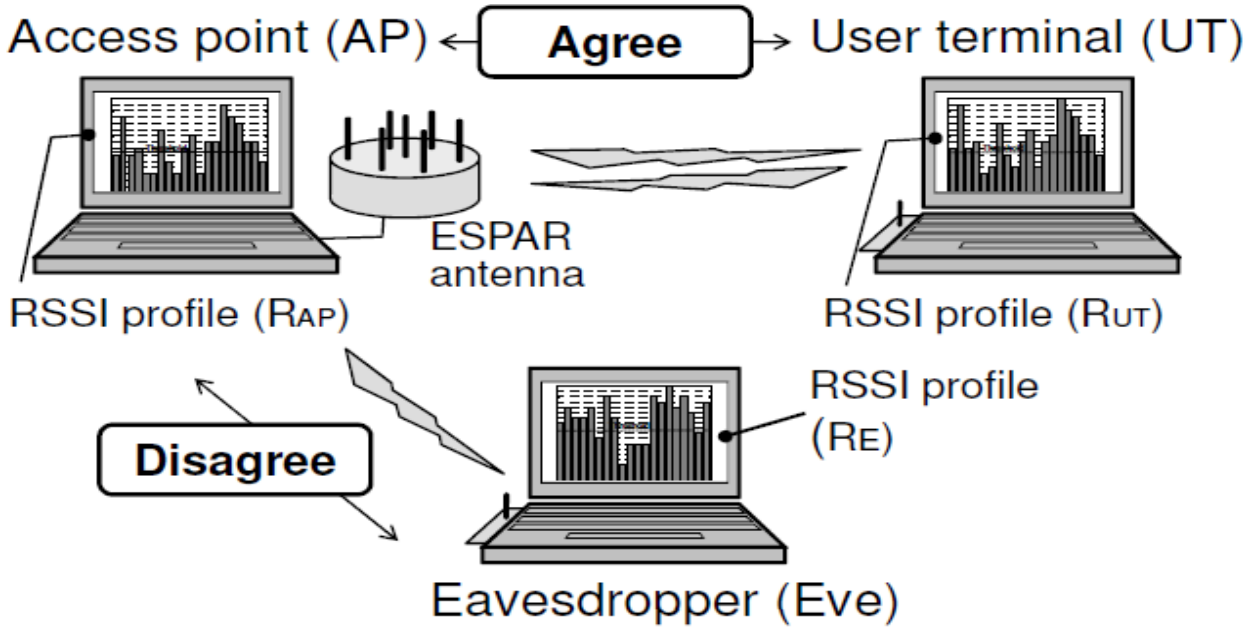
- The communication channel for a communication among two nodes is spatially sharp concentrated[1]
- This channel symmetry can be exploited to derive secure keys among two devices

[1] Smith, A direct derivation of a single-antenna reciprocity relation for the time domain, IEEE Transactions on Antennas and propagation, Vol. 52, no. 6, 2004.

Features of the RF channel

Secure communication based on RSSI measurements[2] [3]

- Utilisation of a variable directional antenna (ESPAR)
  - Increases the fluctuation of channel characteristics based on relative location



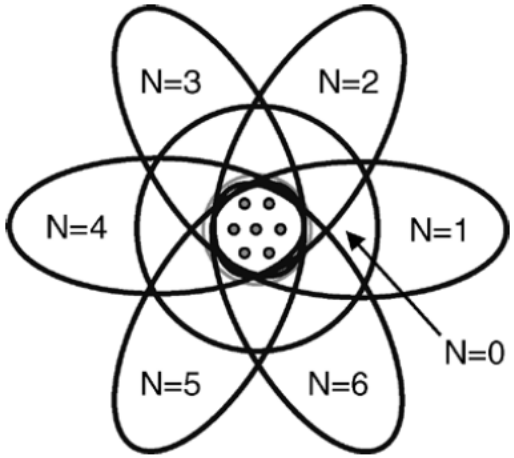
[2] Yasukawa, Iwai, Sasaoka, A secret key agreement scheme with multi-level quantisation and parity check using fluctuation of radio channel property, ISIT, 2008

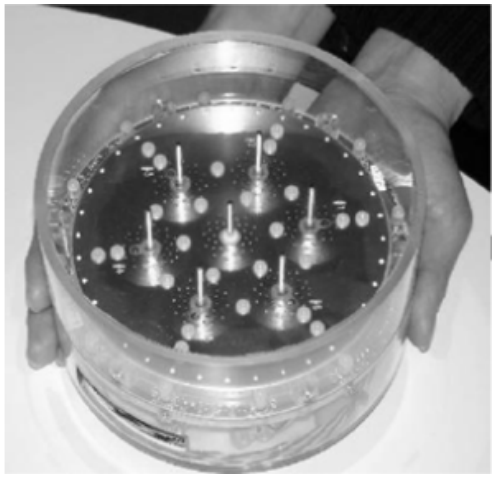
[3] Aono, Higuchi, Ohira, Komiyama, Sasaoka, Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels, IEEE Transactions on Antennas and Propagation, Vol. 53, No. 11, 2005.

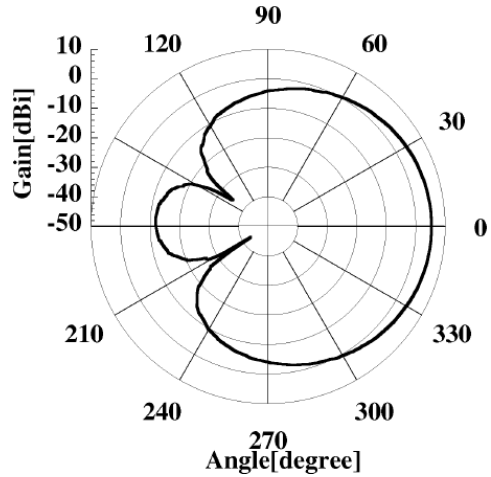
Features of the RF channel

Secure communication based on RSSI measurements

- Utilisation of a variable directional antenna (ESPAR)
  - Variable-directional array antenna
  - Single central active radiator
  - parasitic elements loaded with variable reactors
  - By altering the dc voltage to reactor diodes in the parasitic elements, antenna beam can be formed



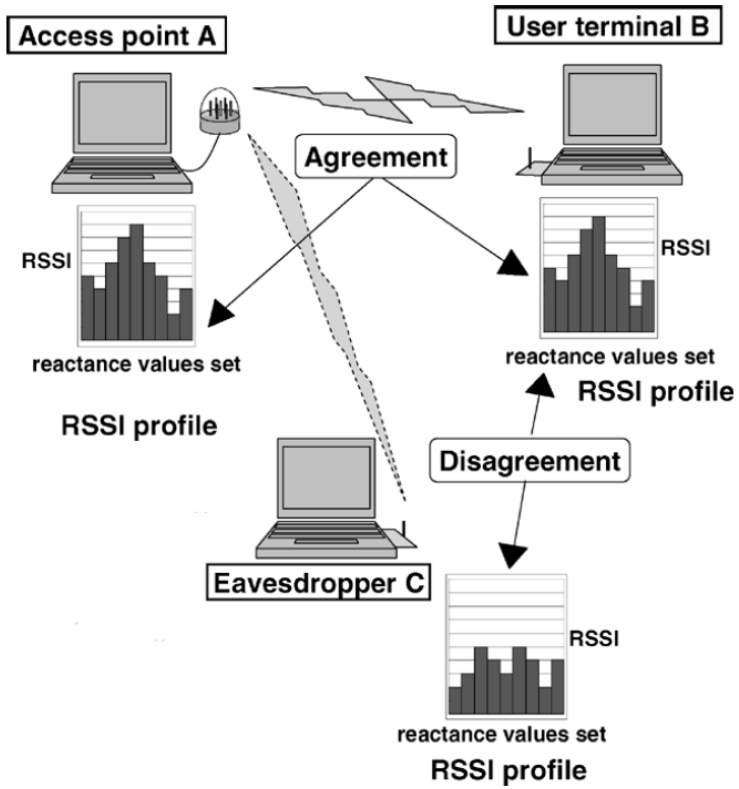




Features of the RF channel

Secure communication based on RSSI measurements

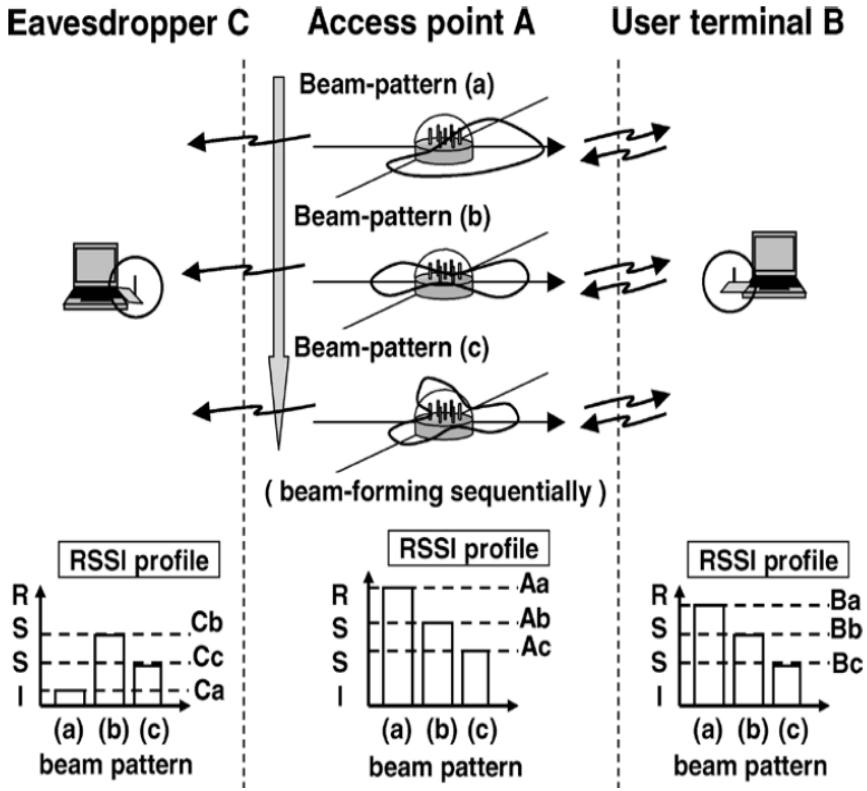
- Secret-key generation and agreement principle
  - Repeated transmission of beam patterns
  - Due to the ESPAR antenna, channel characteristics to spatially separated nodes differ
  - Binary keys are created from the RSSI-sequence according to a threshold value



Features of the RF channel

Secure communication based on RSSI measurements

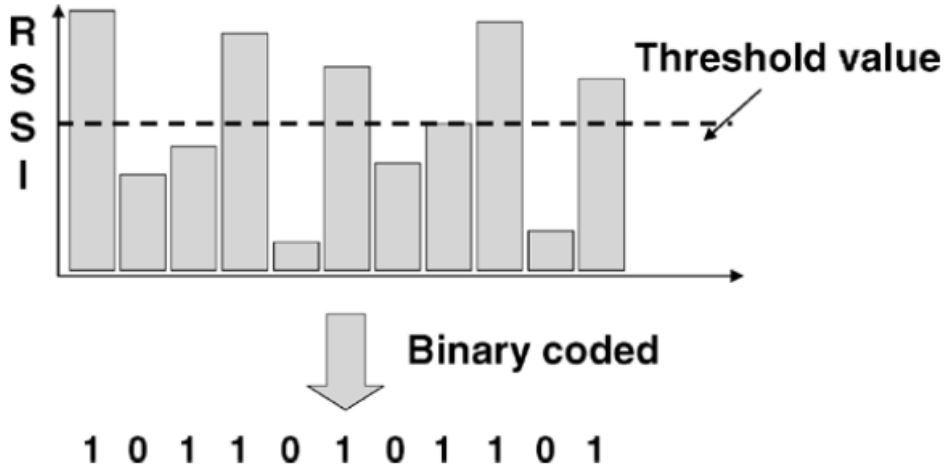
- Secret-key generation and agreement principle
  - Repeated transmission of beam patterns
  - Due to the ESPAR antenna, channel characteristics to spatially separated nodes differ
  - Binary keys are created from the RSSI-sequence according to a threshold value



Features of the RF channel

Secure communication based on RSSI measurements

- Secret-key generation and agreement principle
  - Repeated transmission of beam patterns
  - Due to the ESPAR antenna, channel characteristics to spatially separated nodes differ
  - Binary keys are created from the RSSI-sequence according to a threshold value



Features of the RF channel

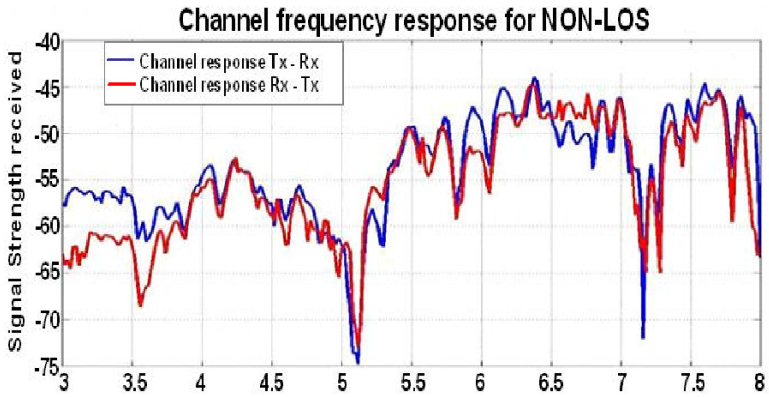
Secure communication based on RSSI measurements

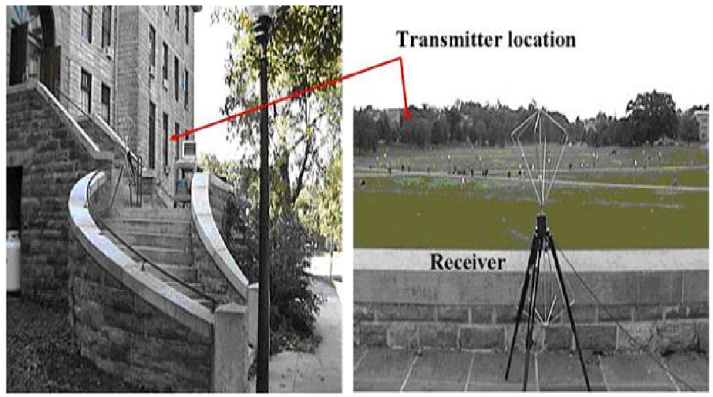
- Discussion
  - Special antenna required to increase spatial fluctuation of channel characteristics
  - Security measure dependent on channel fluctuations

Features of the RF channel

Secure communication based on deep fades in the SNR[4]

- Communication partners agree on a threshold value
- Both nodes transmit repeatedly and alternately
- Channel characteristics are transformed to bit sequence
  - Signal envelope below threshold in timeslot: 1, else 0
- No specialised hardware required
  - Only threshold detectors which are already present in transceivers



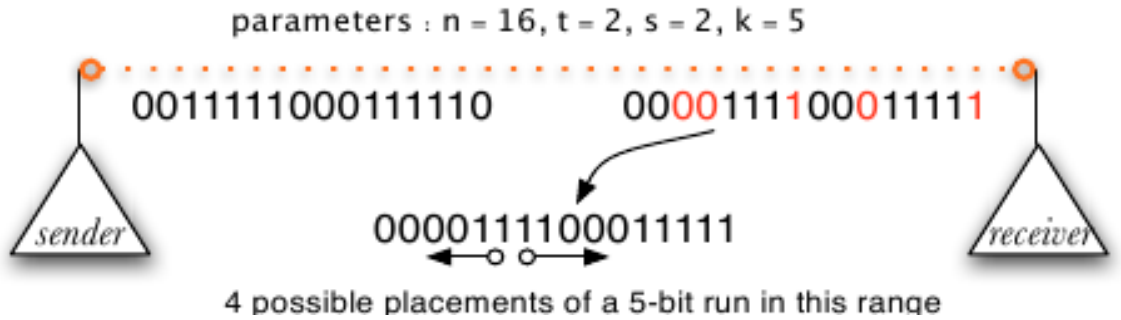


[4] Azimi-Sadjadi, Kiayias, Mercado, Yener, Robust Key Generation from Signal Envelopes in Wireless Networks, CCS, 2007

Features of the RF channel

Secure communication based on deep fades in the SNR

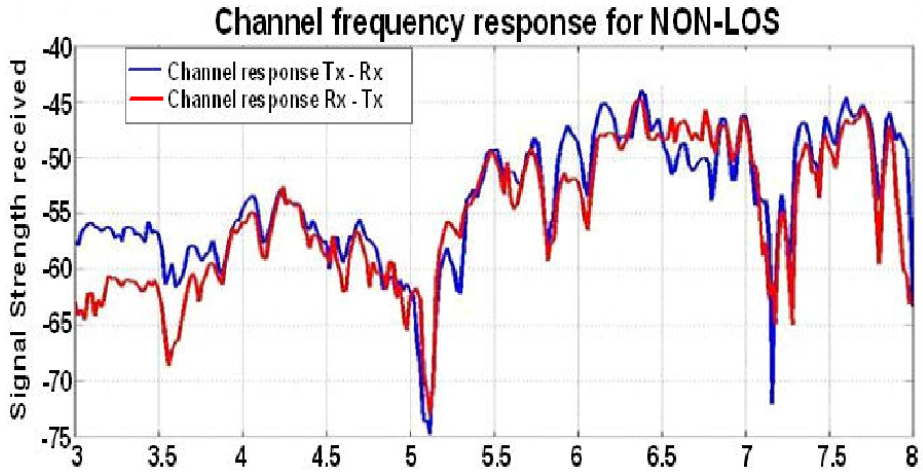
- Key generation
  1. Sender and receiver sample bit sequences
  2. Sender transmits key verification information to receiver
  3. Receiver decides on correct key by scanning through all possible error vectors



Features of the RF channel

Secure communication based on deep fades in the SNR

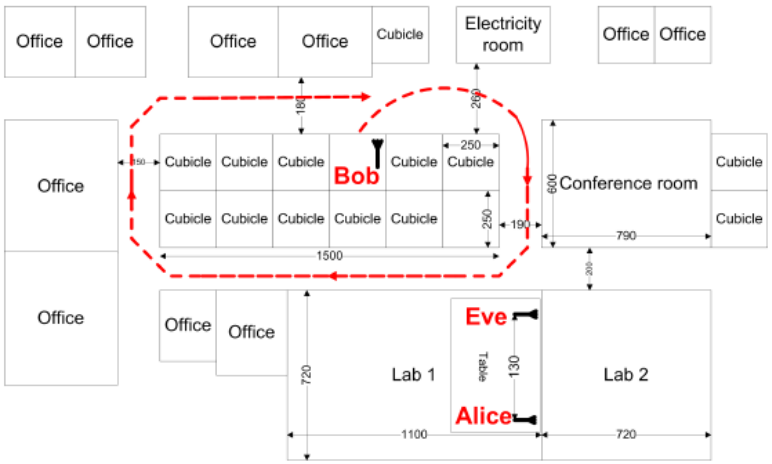
- Discussion
  1. Computationally cheap approach
  2. No special hardware required
  3. Probably uneven distribution of 0 and 1 (Dependent on Channel characteristics and time slot)
  4. Key generation in the presence of noise not optimal

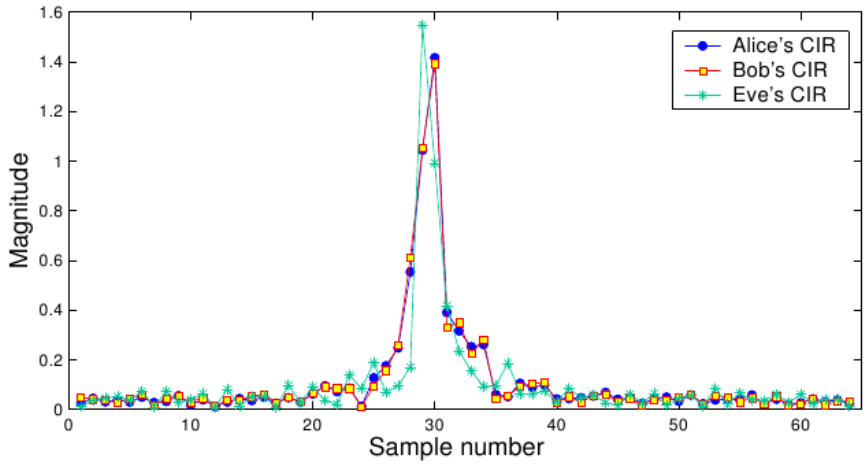


Features of the RF channel

Secure communication based on the CIR[5] [6]

- Utilise Channel impulse response as secure secret
  - Utilise magnitude of CIR pain peak
  - Transformed to binary sequence via Threshold
  - Error correction method required in order to account for noise in the binary sequences



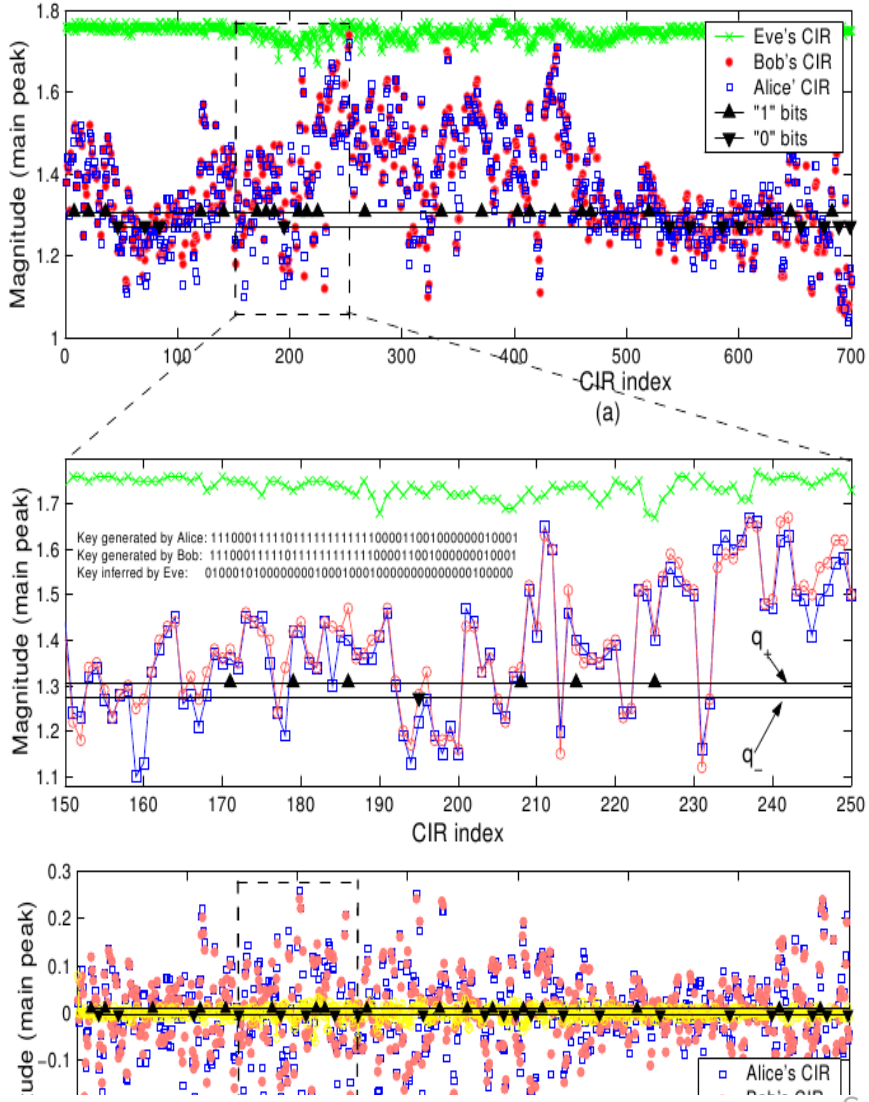


[5] Mathur, Trappe, Mandayam, Ye, Reznik, Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel, MobiCom, 2008

[6] Tmar, Hamida, Pierrot, Castelluccia, An adaptive quantisation algorithm for secret key generation using radio channel measurements, NTMS, 2009

Features of the RF channel

Secure communication based on the CIR



Outline

Fuzzy cryptography

Utilise noise to improve security

Virtually all presently used cryptosystems can theoretically be broken

- by an exhaustive key-search
- Probably, they might even be broken due to novel algorithms
- Or by progress in Computer engineering

By exploring the fact that certain communication channels are inherently noisy, we can achieve secure encryption against adversaries with unbounded computing power

Fuzzy cryptography

Utilise noise to improve security

The security of essentially all presently used cryptosystem is based on at least two assumptions:

1. The computing resources of the adversary are bounded
2. The computational problem of breaking the cryptosystem is computationally infeasible

Both assumption are essentially not proven

1. The model of computation might even be unclear (recently demonstrated by quantum computers which are believed to be more powerful than classical computers)
2. Yet, no lower bound for the hardness of meaningful computational problems

Fuzzy cryptography

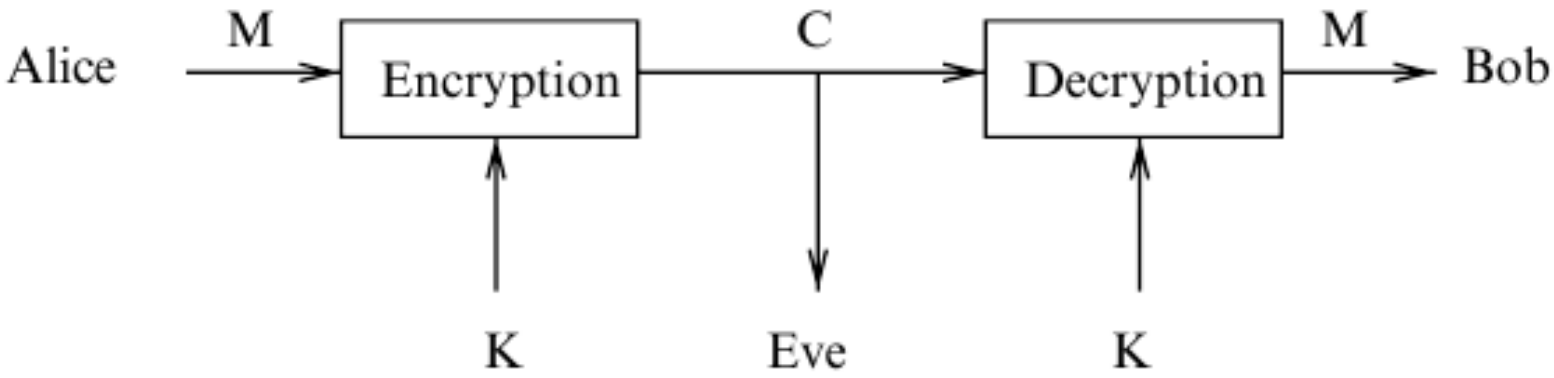
Utilise noise to improve security

Some unconditionally secure cryptosystems are proposed (secure against adversary with unbounded computing power)

Example: One-time pad

- Message $M = [m_1, m_2, \ldots, m_N]$
- Key $K = [k_1, k_2, \ldots, k_N]$ (uniformly distributed N-bit string)
- Cipher-text $C = [c_1, c_2, \ldots, c_N] = [m_1 \oplus k_1, \ldots, m_N \oplus k_N]$

The one-time pad is perfectly secret



Fuzzy cryptography

Utilise noise to improve security

The price we have to pay for perfect secrecy is that

- communicating parties must share a secret key that is at least as long as the message
- and which can only be used once

The scheme is therefore quite impractical

However, Shannon showed that perfect secrecy can not be obtained in a less expensive way

The one-time pad is optimal with respect to key length

Fuzzy cryptography

Utilise noise to improve security

Consequently:

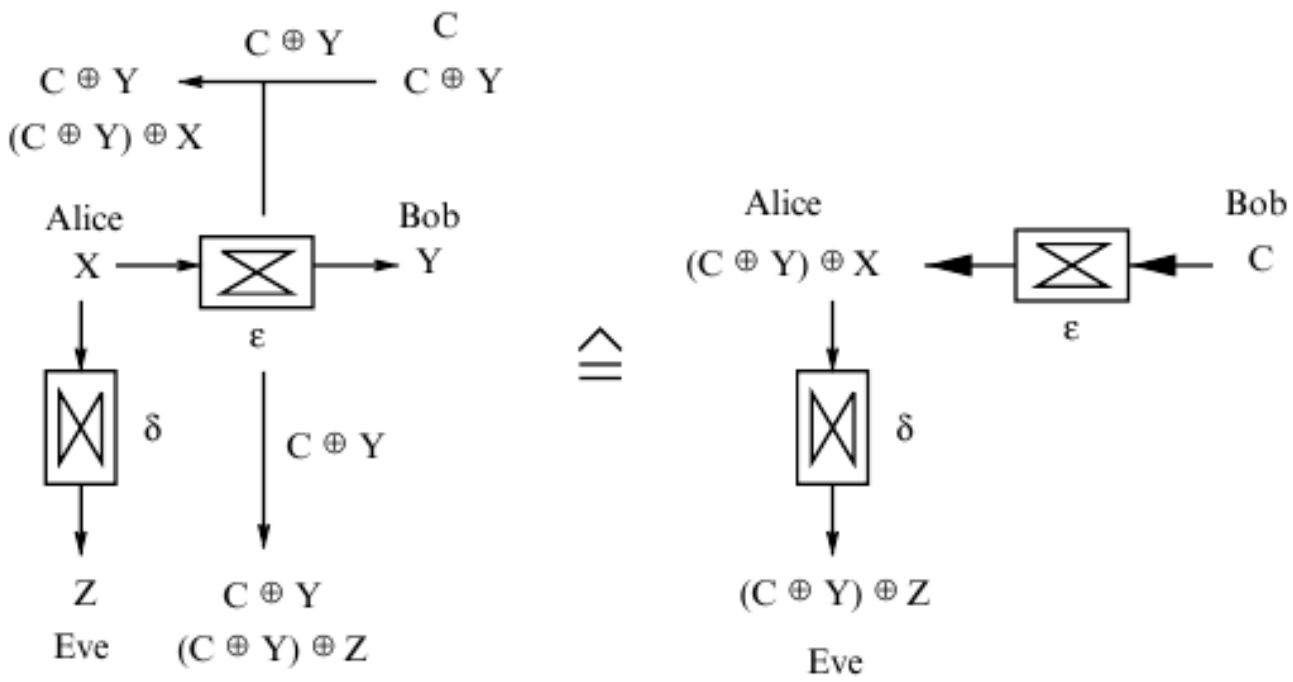
- Every perfectly secret cipher is necessarily as impractical as the one-time pad

However:

- The assumption that the adversary has perfect access to the cipher-text is unrealistic in general
- Every transmission of a signal over a physical channel is subject to noise
- We can utilise noise to achieve a perfectly secure communication at less cost

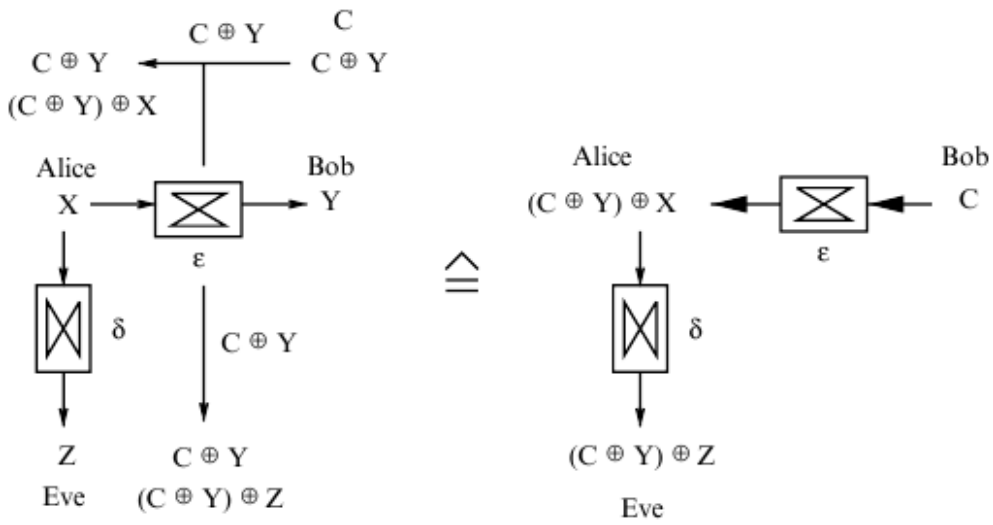
Fuzzy cryptography

Utilise noise to improve security



Fuzzy cryptography

Utilise noise to improve security



By inverting the direction of communication the noise in Eve's reception is increased above those in Alice's

Establishing of a secure key is possible over binary symmetric channel iff the noise in the reception of Eve's message is higher[7]

[7] Wyner, The wire-tap channel, Bell system Technical Journal, 54:1355-1387,1975

Outline

Fuzzy cryptography

Fuzzy Commitment

Traditional cryptographic systems rely on secret bit-strings.

When key contains errors (e.g. noise or mistake), decryption fails.

Rigid reliance on perfectly matching secret keys makes classical cryptographic systems less practicable in noisy systems.

Fuzzy commitment: cryptographic primitive to handle independent random corruptions of bits in a key.

Fuzzy cryptography

Fuzzy Commitment

Traditional cryptographic systems rely on secret bit-strings for secure management of data.

A cryptographic commitment scheme is a function

$$G : C \times X \to Y$$

To commit a value $\kappa \in C$ a witness $x \in X$ is chosen uniformly at random and $y = G(\kappa, x)$ is computed.

A decommitment function takes y and a witness to obtain the original κ

$$G^{-1} : Y \times X \to C$$

Fuzzy cryptography

Fuzzy Commitment

A well defined commitment scheme shall have two basic properties.

Binding It is infeasible to de-commit y under a pair (κ', x') such that $\kappa \neq \kappa'$

Hiding Given y alone, it is infeasible to compute κ

Fuzzy cryptography

Fuzzy Commitment

Fuzzy commitment is an encryption scheme that allows for the use of *approximate* witnesses

Given a commitment $y = G(\kappa, x)$, the system can recover κ from any witness x' that is close to but not necessarily equal to x.

Closeness in fuzzy commitment is measured by Hamming distance.

Fuzzy cryptography

Fuzzy Commitment

A fuzzy commitment scheme may be based on any (linear) error-correcting code

An error-correcting code consists of

Message space $M \subseteq F^a$ (F^i denotes all strings of length i from a finite set of symbols F)

Codeword space $C \subseteq F^b$ with $(b > a)$

Bijection $\theta : M \leftrightarrow C$

Decoding function $f : C' \rightarrow C \cup \perp$ (The symbol $\perp$ denotes the failure of f)

The function f maps an element in C' to its nearest codeword in C.

Fuzzy cryptography

Fuzzy Commitment

Noise of physical function may be viewed as the difference $c - c'$

Decoding function f applied to recover original codeword c

This is successful if c' is close to c. In this case: $c = f(c')$

The minimum distance of the code is the smallest distance $d = Ham(c - c')$ between any two codewords $c, c' \in C$

Typically, it is possible to correct at least $\frac{d}{2}$ errors in a codeword

Fuzzy cryptography

Fuzzy Commitment

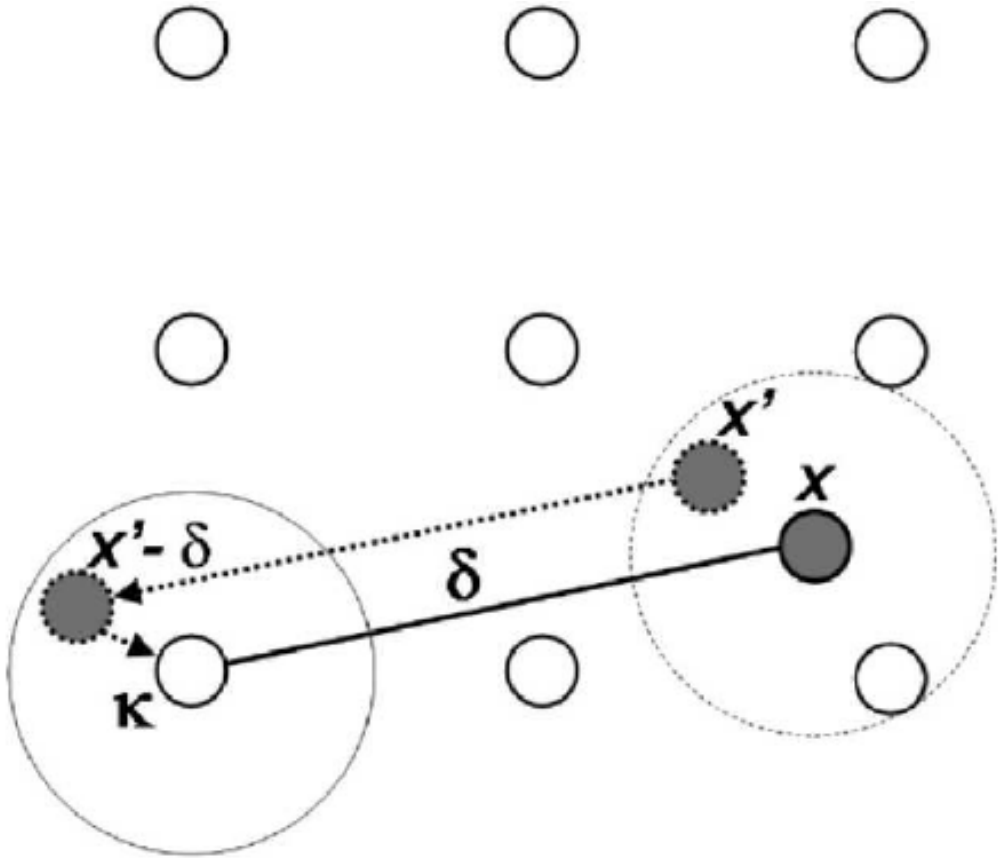
For fuzzy commitment, the secret key κ is chosen uniformly at random from the codeword space C. Then,

1. An offset $\delta = x - \kappa$ is computed
2. A one-way, collision-resistant hash function is applied to obtain $h(\kappa)$
3. $y = (\delta, h(\kappa))$ is made public
4. $\kappa' = f(x' - \delta)$ is computed
5. It is possible to de-commit y under a witness x' with $Ham(x, x') < \frac{d}{2}$

Once κ is recovered, its correctness my be verified by computing $z = h(\kappa)$

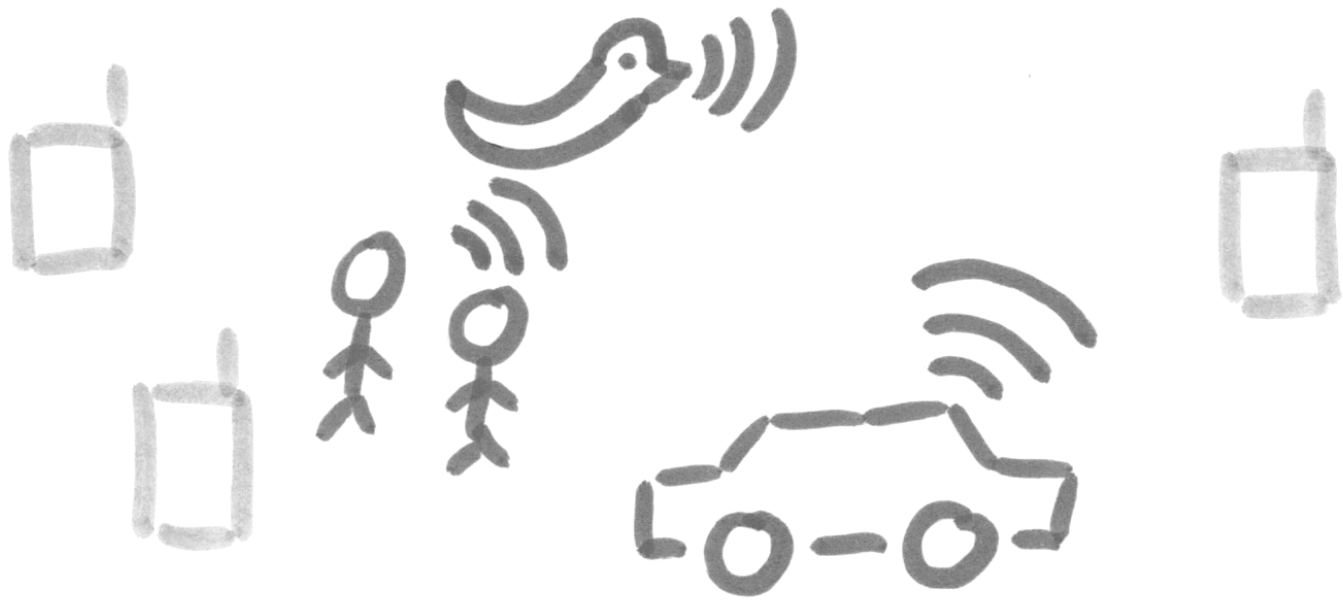
Fuzzy cryptography

Fuzzy Commitment

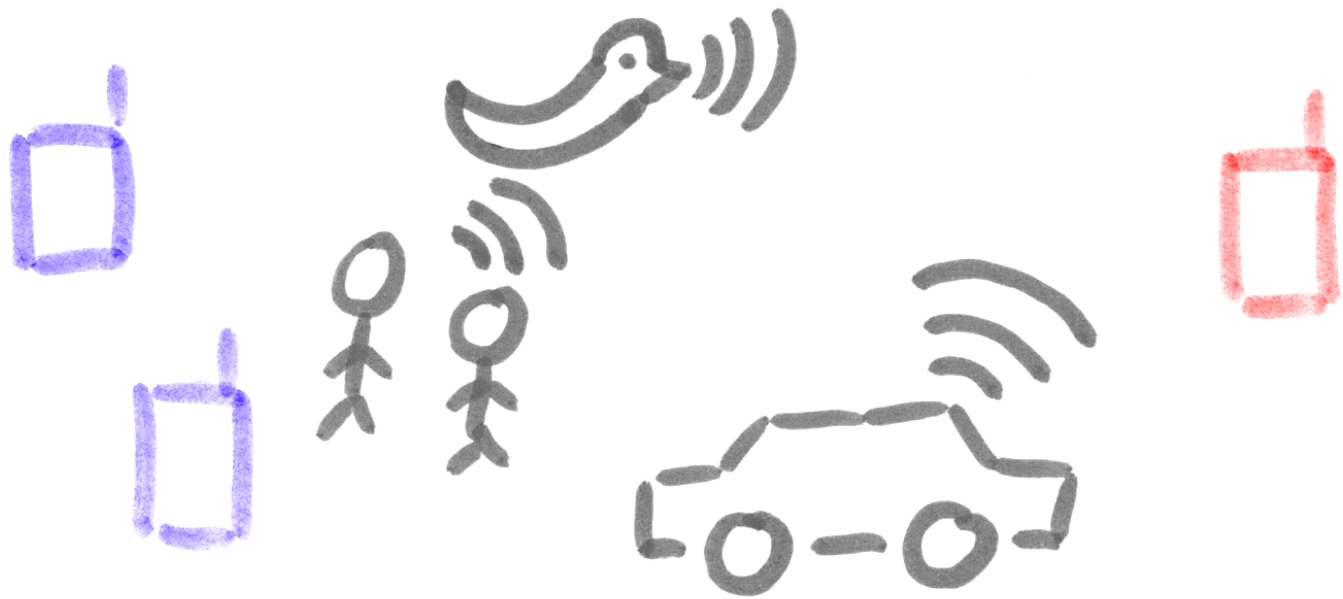


Spontaneous audio-based device pairing

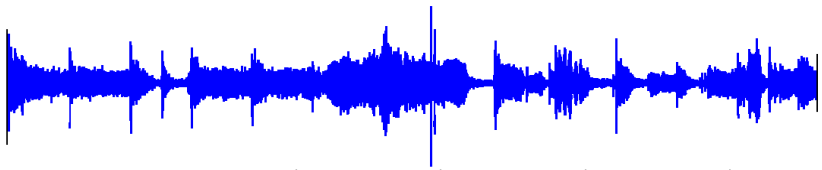
Spontaneous audio-based device pairing



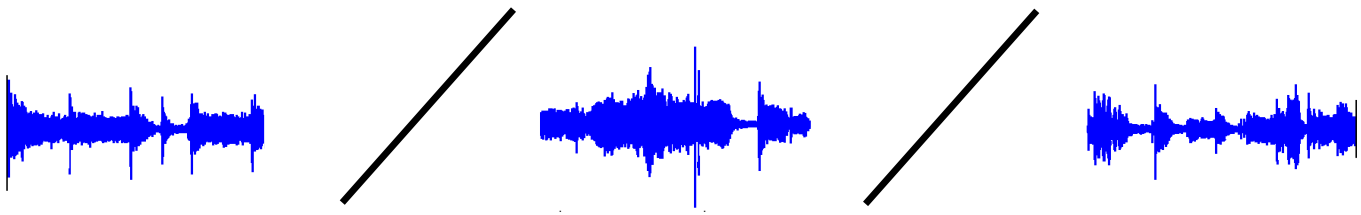
Spontaneous audio-based device pairing



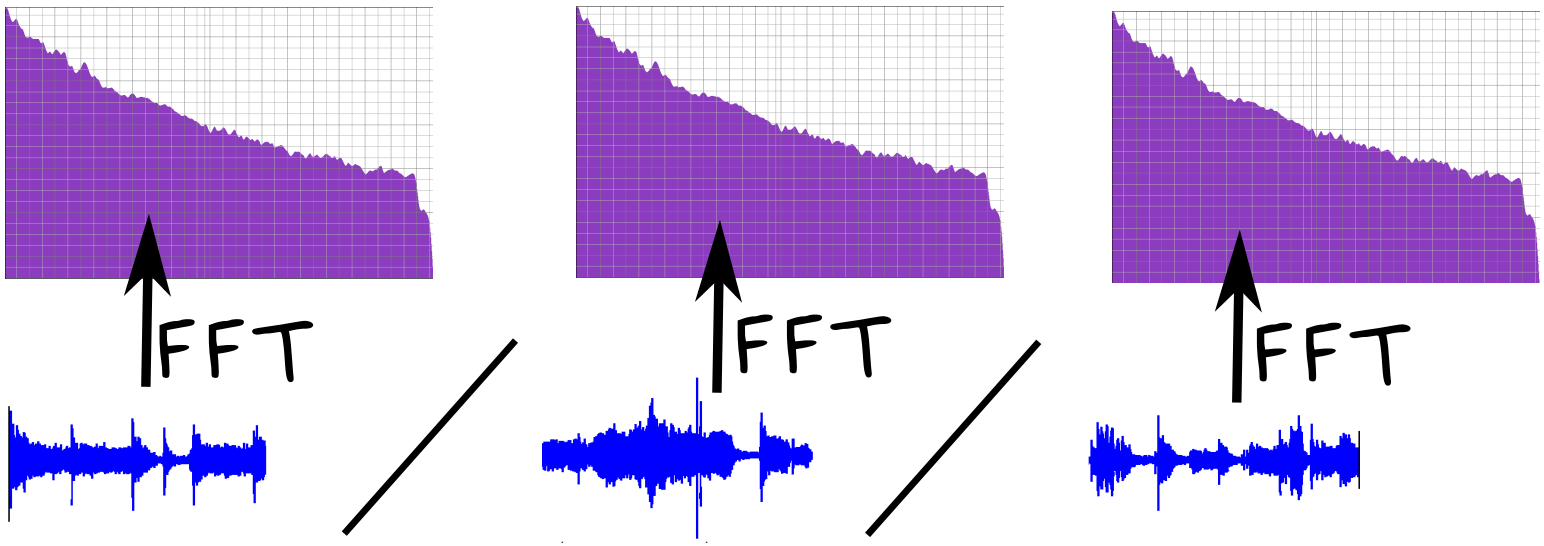
Spontaneous audio-based device pairing



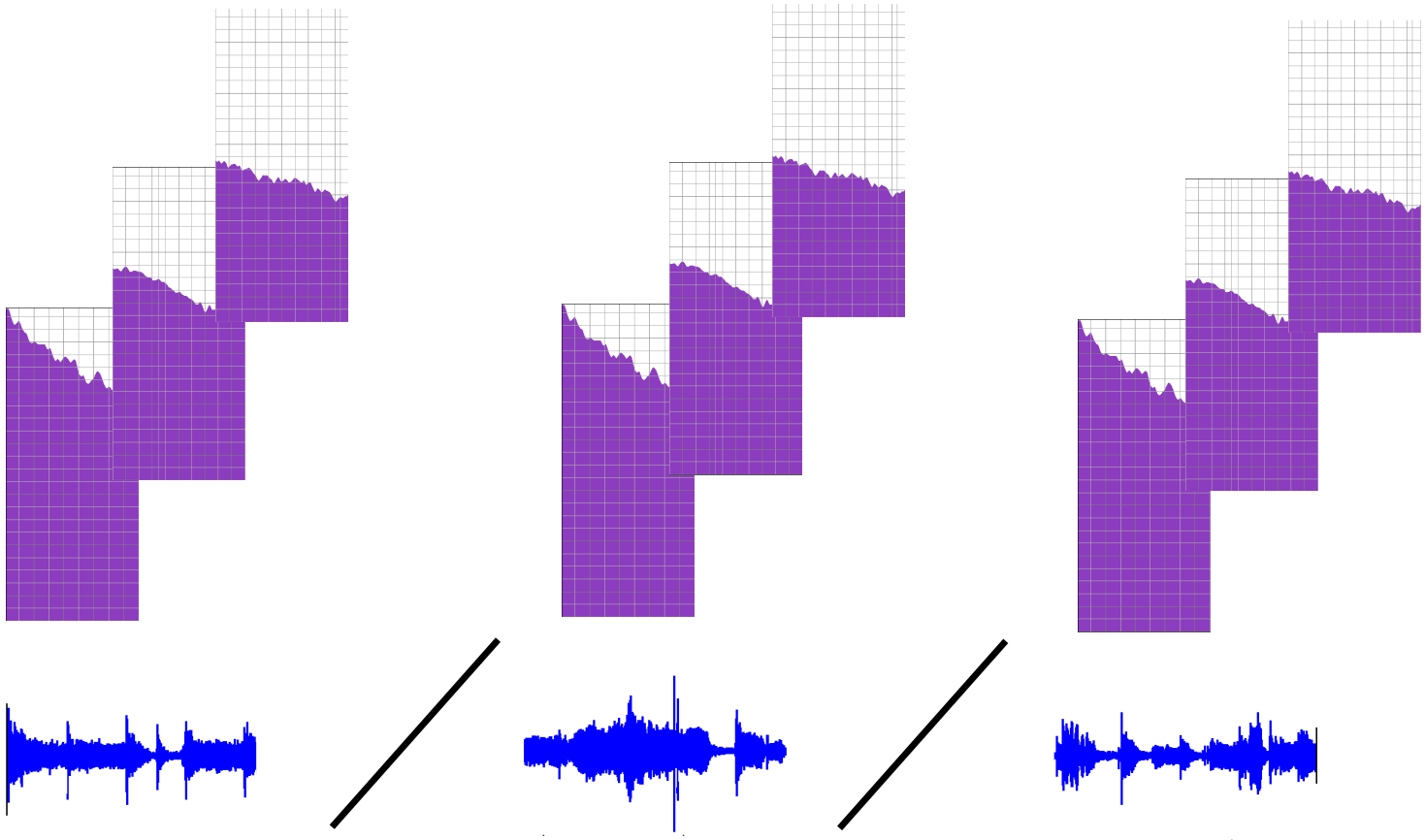
Spontaneous audio-based device pairing



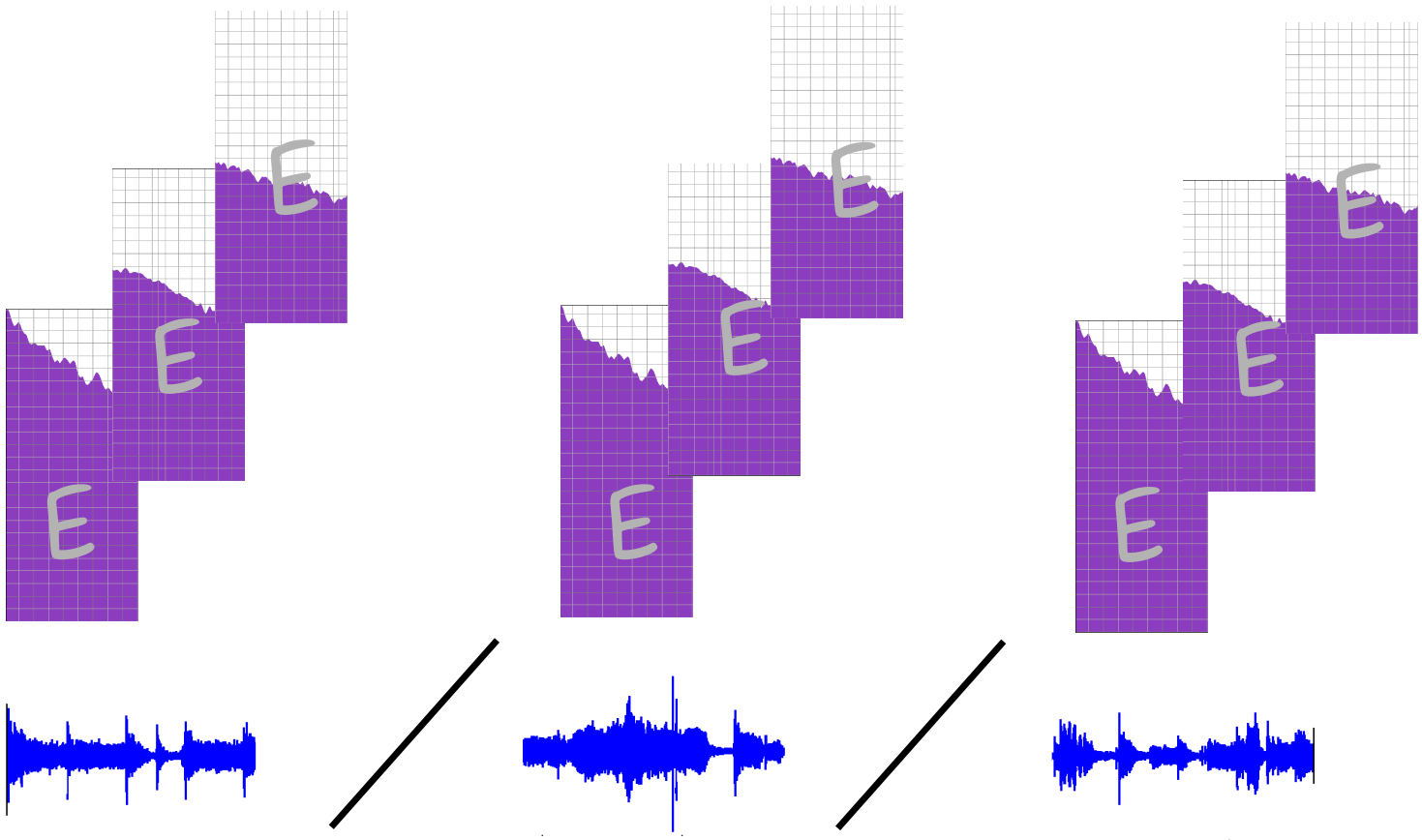
Spontaneous audio-based device pairing



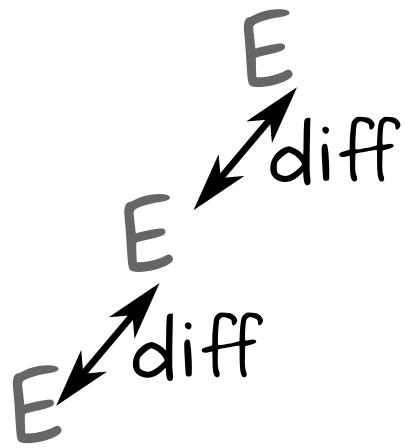
Spontaneous audio-based device pairing

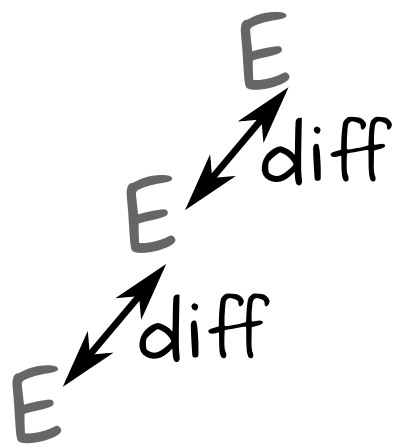


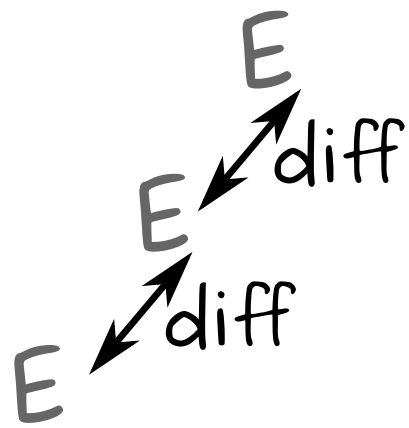
Spontaneous audio-based device pairing



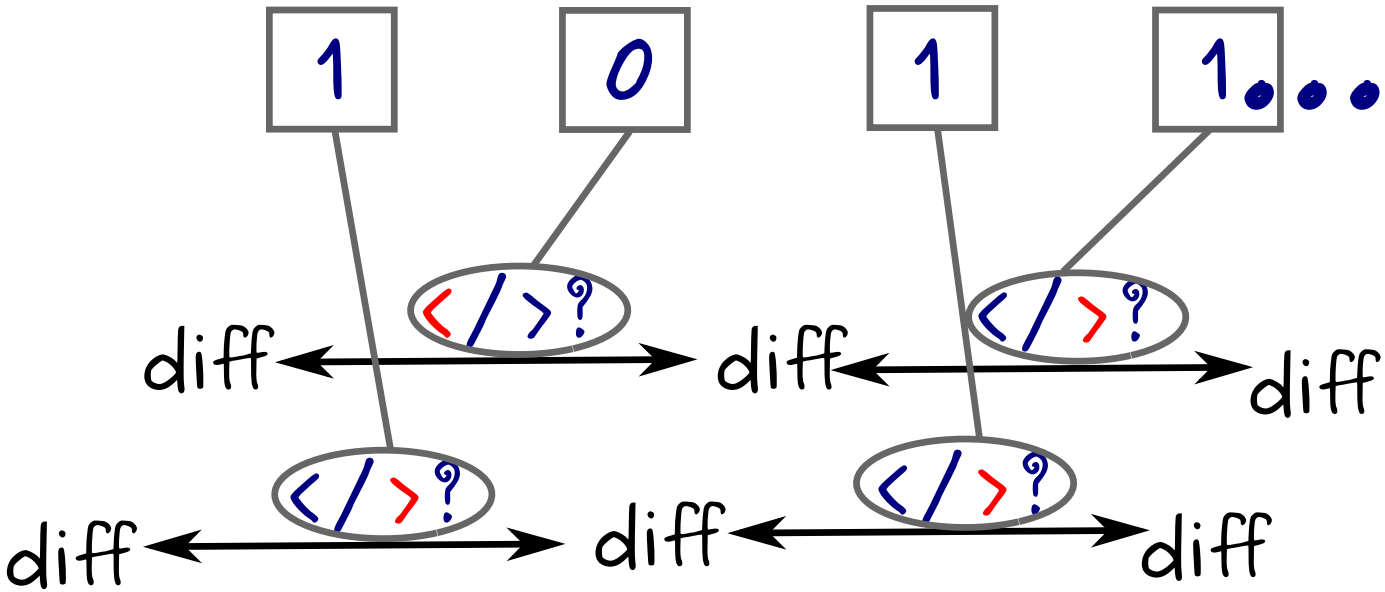
Spontaneous audio-based device pairing





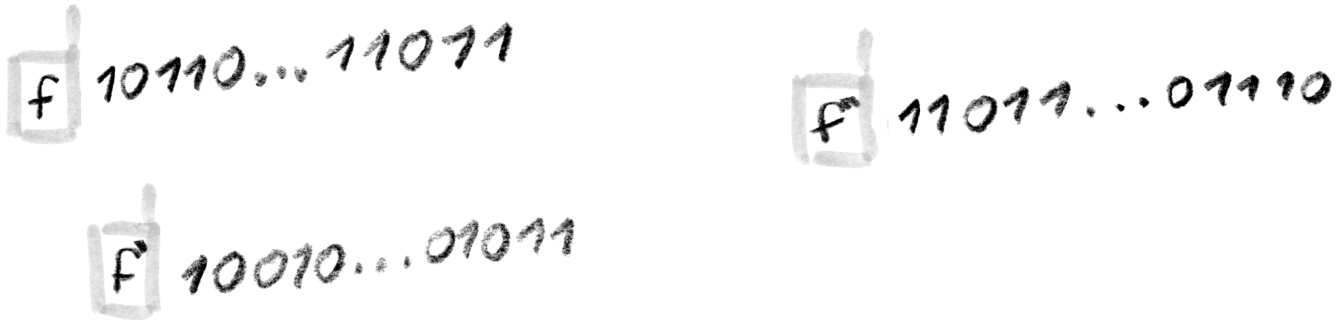


Spontaneous audio-based device pairing

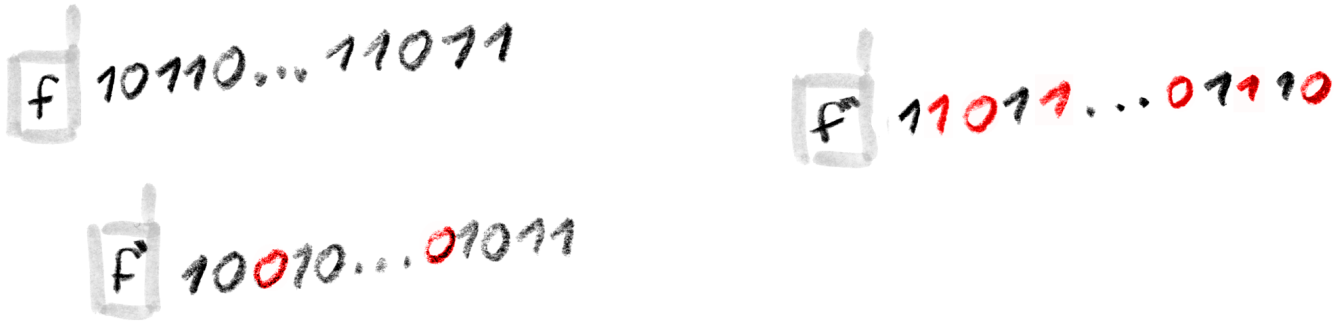


Spontaneous audio-based device pairing

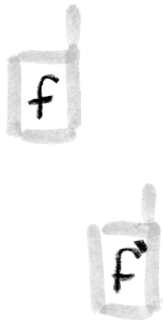
Spontaneous audio-based device pairing



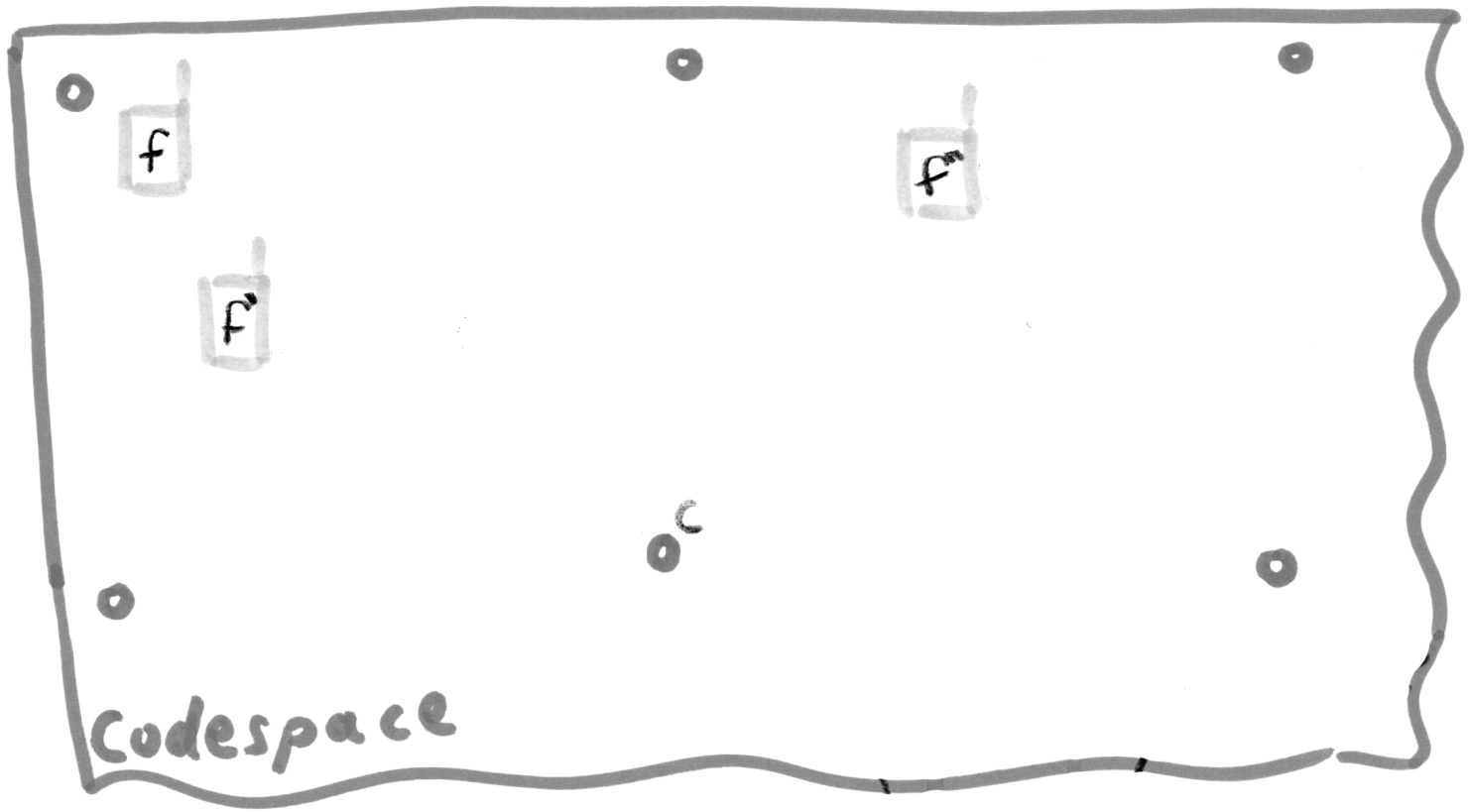
Spontaneous audio-based device pairing



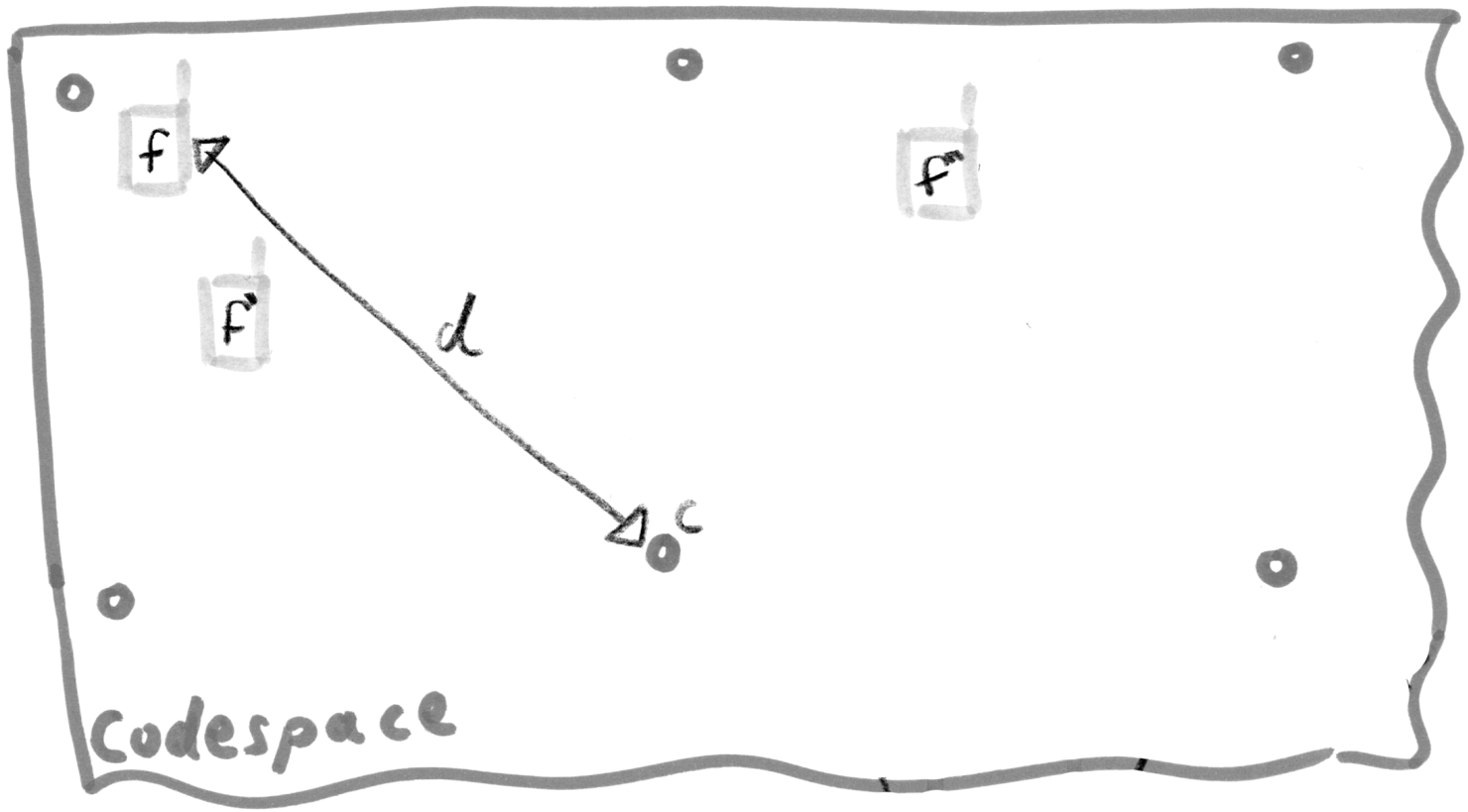
Spontaneous audio-based device pairing



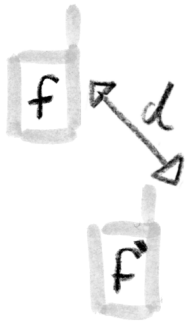
Spontaneous audio-based device pairing



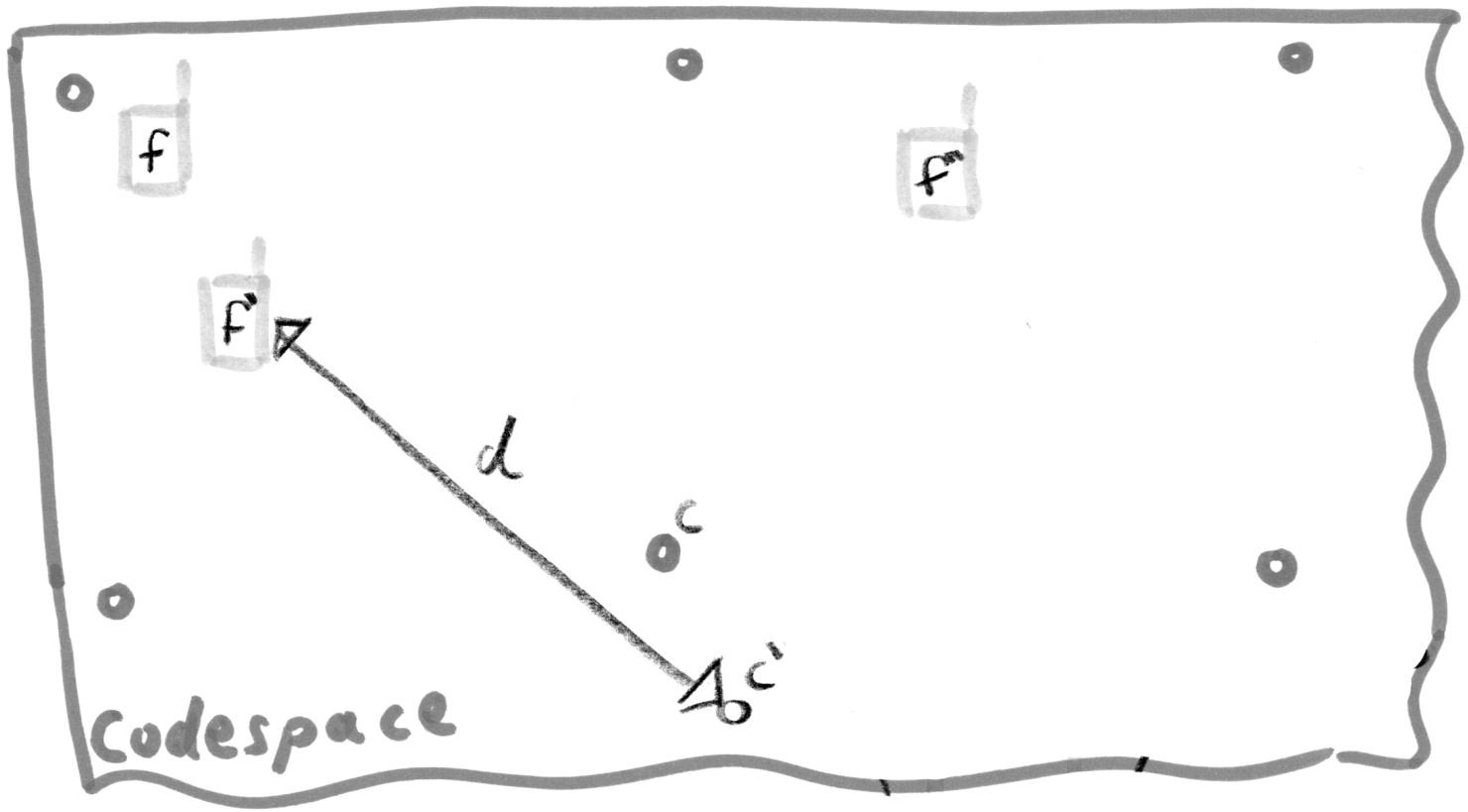
Spontaneous audio-based device pairing



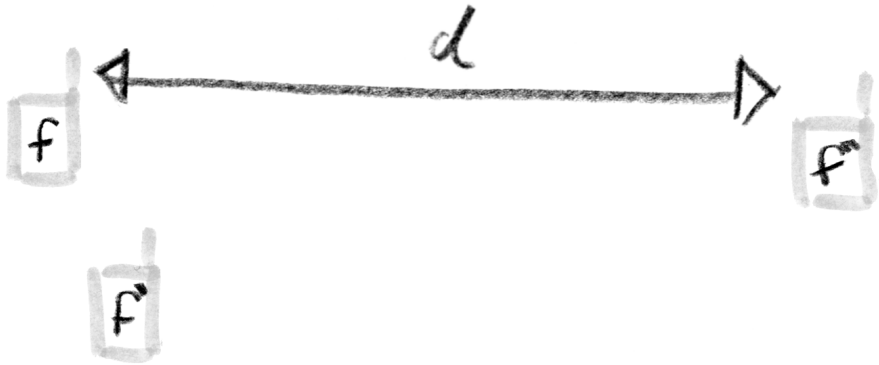
Spontaneous audio-based device pairing



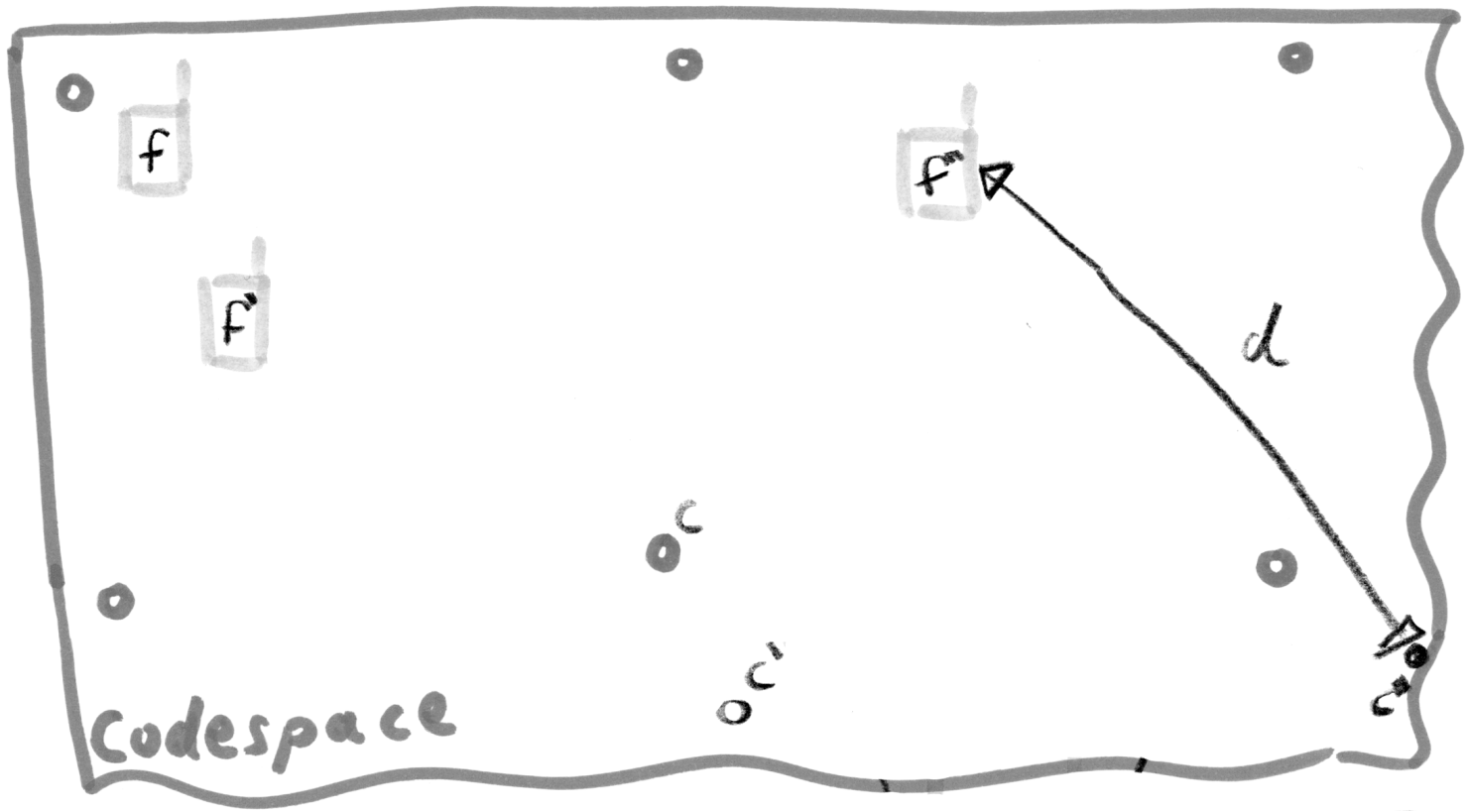
Spontaneous audio-based device pairing



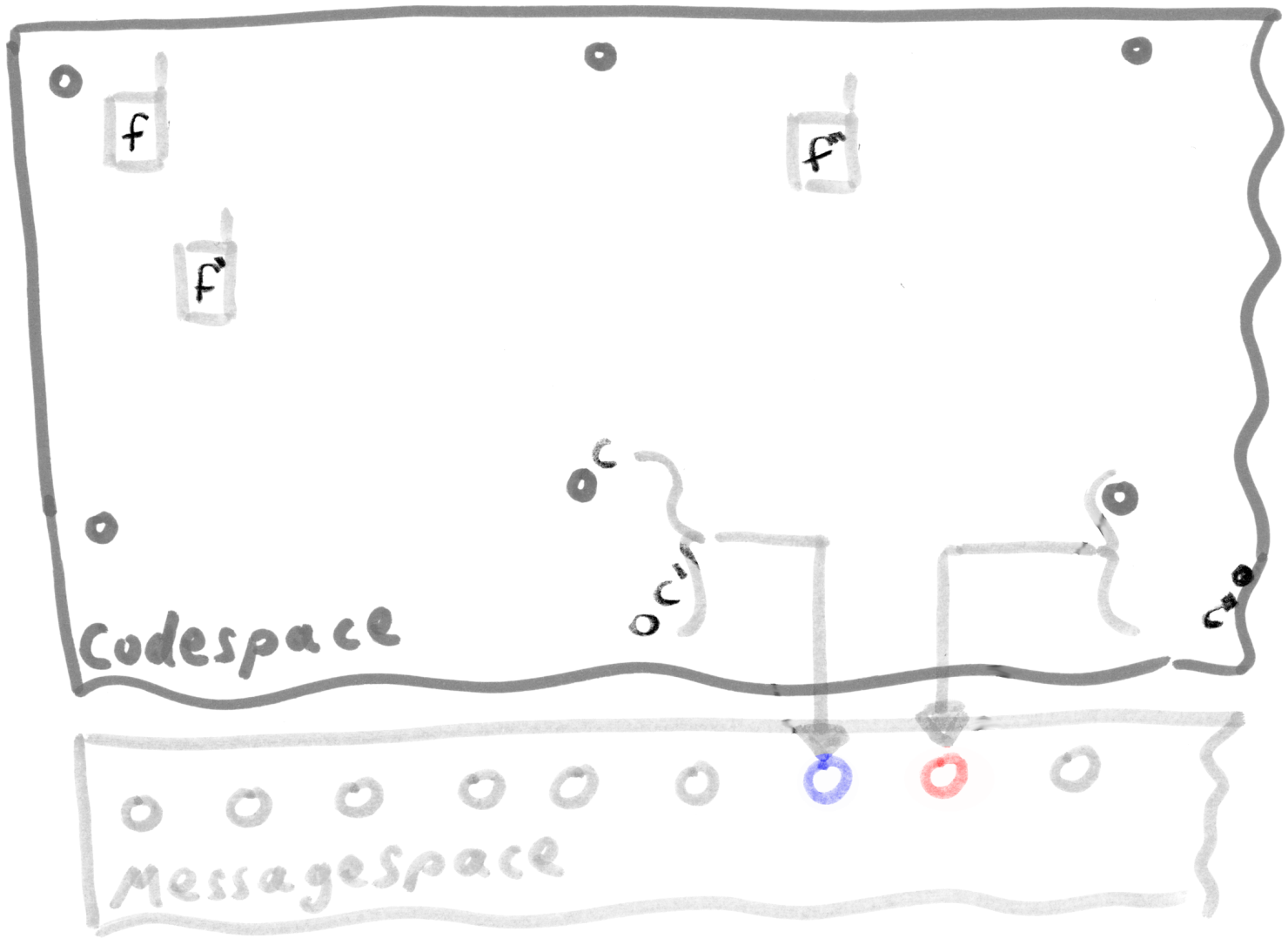
Spontaneous audio-based device pairing



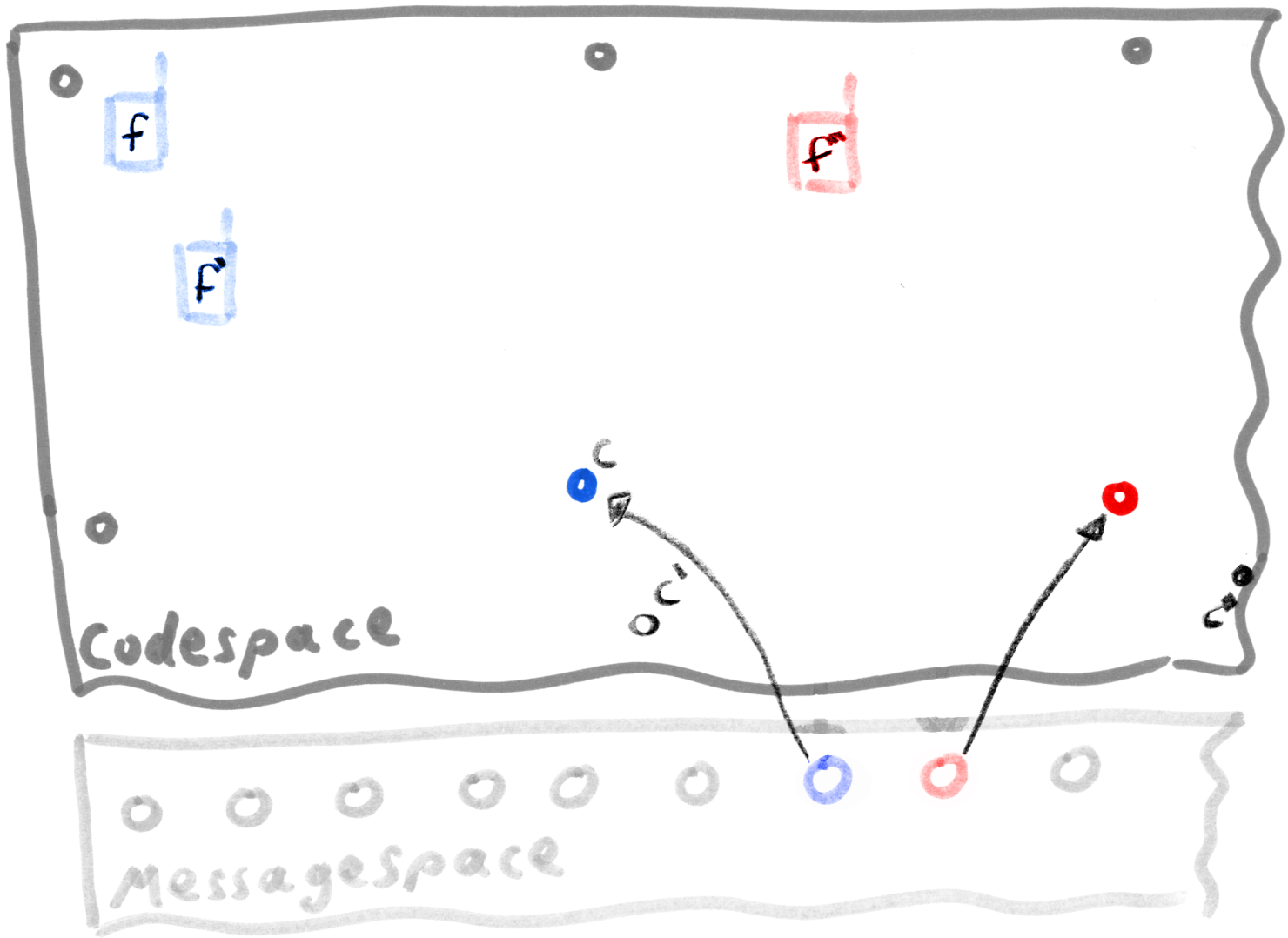
Spontaneous audio-based device pairing



Spontaneous audio-based device pairing



Spontaneous audio-based device pairing



Outline

Block codes

Let q denote the number of distinct symbols employed on the channel

Block code is a set of M sequences of channel symbols of length n

Decision to which code word a received word belongs may be based on a decoding table

Code Words	1 1 0 0 0	0 0 1 1 0	1 0 0 1 1	0 1 1 0 1
	1 1 0 0 1	0 0 1 1 1	1 0 0 1 0	0 1 1 0 0
	1 1 0 1 0	0 0 1 0 0	1 0 0 0 1	0 1 1 1 1
Other	1 1 1 0 0	0 0 0 1 0	1 0 1 1 1	0 1 0 0 1
Received	1 0 0 0 0	0 1 1 1 0	1 1 0 1 1	0 0 1 0 1
Words	0 1 0 0 0	1 0 1 1 0	0 0 0 1 1	1 1 1 0 1
	$\overline{1}$ $\overline{1}$ $\overline{1}$ $\overline{1}$ $\overline{0}$	$\overline{0}$ $\overline{0}$ $\overline{0}$ $\overline{0}$ $\overline{0}$	$\overline{0}$ $\overline{1}$ $\overline{0}$ $\overline{1}$ $\overline{1}$	$\overline{1}$ $\overline{0}$ $\overline{1}$ $\overline{0}$ $\overline{1}$
	0 1 0 1 0	1 0 1 0 0	1 1 1 1 1	0 0 0 0 1

Block codes

Linear block codes

For linear block codes we require a set of k basis vectors $\overrightarrow{g}$ (generator vectors) of length n

Basis vectors are linear independent vectors that span the basis of a vector space

These vectors are considered as rows of a matrix G

The row-space of G defines the linear code V and code vectors $\overrightarrow{v}$ are linear combinations of rows in G

Important: vectors g must be linear independent. Otherwise different linear combinations lead to identical code vectors

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Block codes

Linear block codes

Data vectors $\overrightarrow{d}$ define which generator vectors g are combined to a code vector $\overrightarrow{v}$

We define a matrix H of rank $n-k$ whose row space is a basis of vectors orthogonal to each vector in G (null space)

Since each code vector $\overrightarrow{v}$ is the result of a linear combination of generator vectors $\overrightarrow{g}$, we have

$$\overrightarrow{v}H^T = \overrightarrow{0}$$

In the case of errors in the code vector, the result is hence

$$(\overrightarrow{v} + \overrightarrow{e})H^T \neq \overrightarrow{0}$$

iff $(\overrightarrow{v} + \overrightarrow{e}) \notin \overrightarrow{\alpha}G$

Block codes

Linear block codes

In the case of errors in the code vector, the result is hence

$$(\vec{v} + \vec{e})H^T \neq \vec{0}$$

The error vector $\vec{e}$ then defines the linear combination of rows of H^T that lead to the syndrome:

$$\begin{aligned} & (\vec{v} & + & \vec{e})H^T \\ = & \vec{v}H^T & + & \vec{e}H^T \\ = & \vec{0} & + & \vec{e}H^T \\ = & \vec{s} & & \end{aligned}$$

Since H is spanned by basis vectors: $\vec{s}$ defines uniquely the error vectors.

Outline

Physical random functions

Physical random functions / Physically unclonable functions:

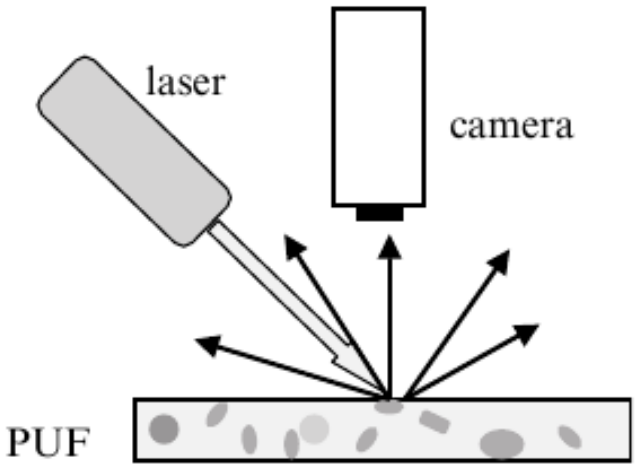
Random functions that can only be evaluated with the help of a physical system

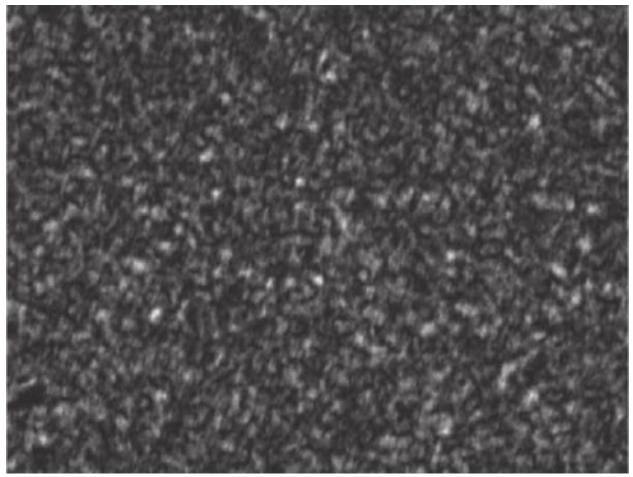
Definition

A PUF is a random function that can only be evaluated with the help of a specific physical system. The inputs to a physical random function are challenges and the outputs are responses.

Physical random functions

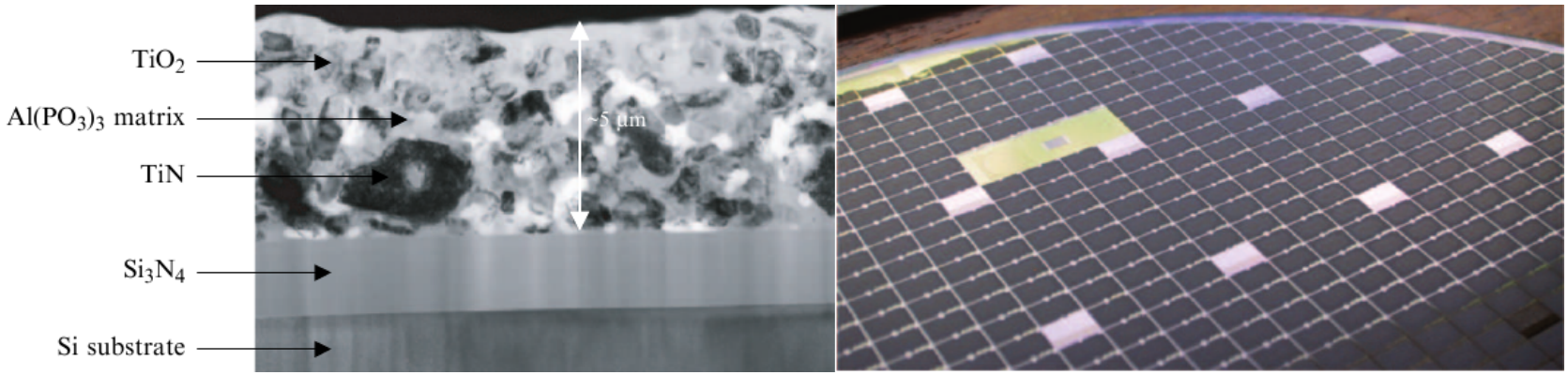
Optical PUFs Made of transparent optical medium containing bubbles. Shining a laser beam through the medium produces speckle pattern (response) that depends on exact position/direction of incoming beam.





Physical random functions

Silicon PUFs From the input to a circuit that reconfigures the path that signals follow through the circuit, an output is related to the time it takes for signals to propagate through a complex circuit.



Physical random functions

Security of PUFs relies on difficulty of extracting all necessary parameters from a complex physical system

Attacker trying to extract all physical parameters might modify the PUF in the process

This makes PUFs tamper resistant to some extend

Physical random functions

PUF implementations build on random manufacturing variations (bubble position or exact wire delays):
Exact behaviour is a mystery even for the manufacturer

Not feasible to create two identical copies of a PUF

A difficulty of optical and silicon PUFs is that their output is noisy

Error correction that does not compromise the security is required[8]

[8] G.E. Suh, C.W. O'Donnell, I. Sachdev, S. Devadas, Design and implementation of the AEGIS single-chip secure processor using physical random functions, Proceedings of the 32nd Annual International Symposium of computer Architecture, 2005

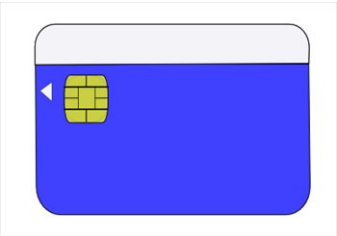
Physical random functions

Standard application: Key-card[9]

Lock stores a database of challenge response pairs (CRPs) for PUF

When the bearer of the PUF wants to open the lock, it selects a challenges it knows and asks the PUF for the corresponding response

Each CRP can be used only once : Card will eventually run out of PUFs



[9] R. Pappu, Physical One-Way Functions, PhD thesis, MIT, 2001

Controlled Physical random functions

Definition

Controlled physical random function (CPUF):
PUF that can only be accessed through specific API

Main problem with uncontrolled PUFs: Anybody can query the PUF for the response to any challenge

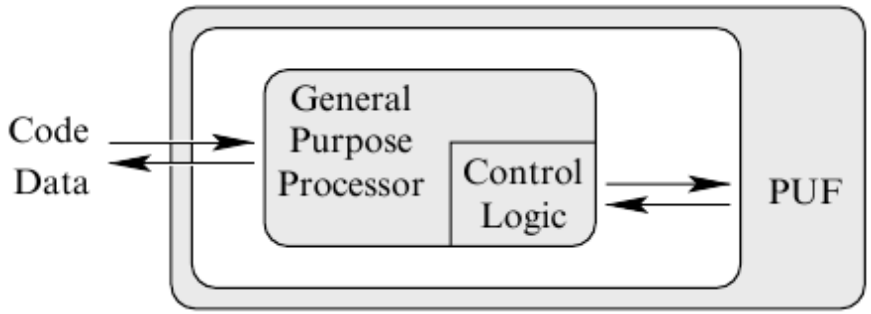
In order to engage in cryptography with a PUF device, a user has to exploit the fact that only he and the device know the response to a specific challenge.

Controlled Physical random functions

Third party could try to overhear challenge, obtain response from PUF and spoof the device

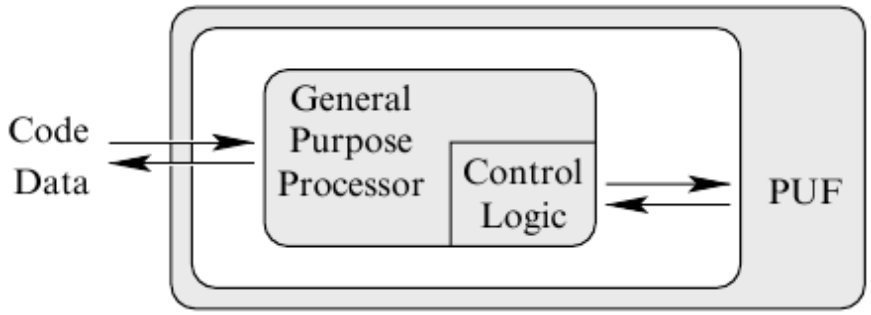
Problem: Adversary can freely query the PUF

By using CPUFs, Access to PUF restricted by control algorithm that prevents this attack



Embedding control logic for PUF in physical system of PUF makes it difficult to conduct invasive attacks on the control logic

Controlled Physical random functions



The PUF and its control logic have complementary roles

The PUF protects the control logic from invasive attacks

The control logic protects the PUF from protocol attacks

Controlled Physical random functions

Applications for CPUFs
Applications for CPUFs include applications that require single symmetric key on a chip

- Smartcards that implement authentication:
  Current smart-cards: Hidden digital keys can be extracted using various attacks

  PUF on the smartcard: Can authenticate chip – Digital key not required (Smartcard hardware itself is the secret key)

  Key can not be duplicated: Person that temporary looses control of card need not fear that an adversary might have cloned the card or that the security became somehow impaired.

Questions?

Stephan Sigg
sigg@nii.ac.jp

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- P. Tulys, B. Skoric, T. Kevenaar: Security with Noisy Data – On private biometrics, secure key storage and anti-counterfeiting, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.