Introduction to Modern Cryptography
Lecture 12

December 31, 2013

Instructor: Benny Chor

Teaching Assistant: Nir Bitansky

School of Computer Science
Tel-Aviv University

Fall Semester, 2013–14, 15:00–18:00
Dan David 201

Course site: http://tau-crypto-f13.wikidot.com/

Lectures 10 and 11: Reminder

- Secret sharing: more details, and a proof.
- Interactive proof systems.
- Zero knowledge proof systems.

- Fully homomorphic encryption (guest lecture by Prof. Zvika Brakerski, WIS)

Lectures 12: Plan

- Secure function evaluation in a multi party setting.
- Oblivious transfer.
- Yao's garbled circuit.

- Secret sharing and error correction codes.

Much of this lecture will be done on the board.

Secure Multi Party Computation

We have n parties $p_1, \ldots, p_n$, each having one input $x_1, \ldots, x_n \in \Sigma$.

Each party values its own privacy, and is suspicious about the intentions of the other parties.

Their goal is to cooperate and compute $f(x_1, \ldots, x_n)$.

What issues may arise?

Secure Multi Party Computation: Many Alternative Models

- Type of adversarial behavior: Honest but curious vs. malicious.
- Number of parties: $n = 2$ vs. $n > 2$.
- Size of adversary coalition t.
- Computationally bounded vs. unbounded participants.
- Cryptographic assumption made (general vs. specific).
- Nature of timing (asynchronous vs. synchronous).

and many other alternatives, not mentioned here.

Example: Independence of Inputs

Consider $f(x, y) = x \oplus y$, where $x, y \in \{0, 1\}$.

The outcome plus any input determines the other input.

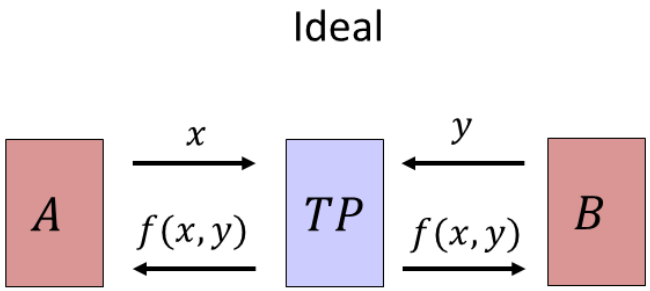
But in some settings want to guarantee independence of inputs.

Secure Multi Party Computation: Plan

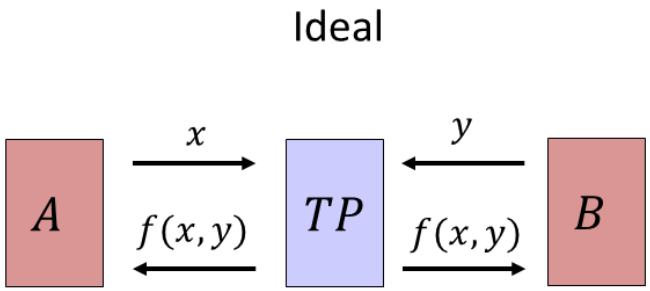
- We will cover just two alternatives:
- Honest but curious participants.
- $n = 2$ and $t = 1$.
- Synchronous timing.
- Computationally bounded vs. computationally unbounded participants.

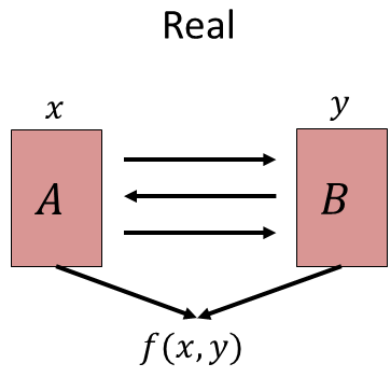
While these choices cover only a small fraction of the spectrum, they provide an exposure to many of the important ideas. Further details are left to more advanced courses, and/or to independent reading from the research literature.

Secure Two Party: Ideal Model vs. Real Model
(figures by an up and coming, but still anonymous, artist)



Secure Two Party: Ideal Model vs. Real Model (figures by an up and coming, but still anonymous, artist)





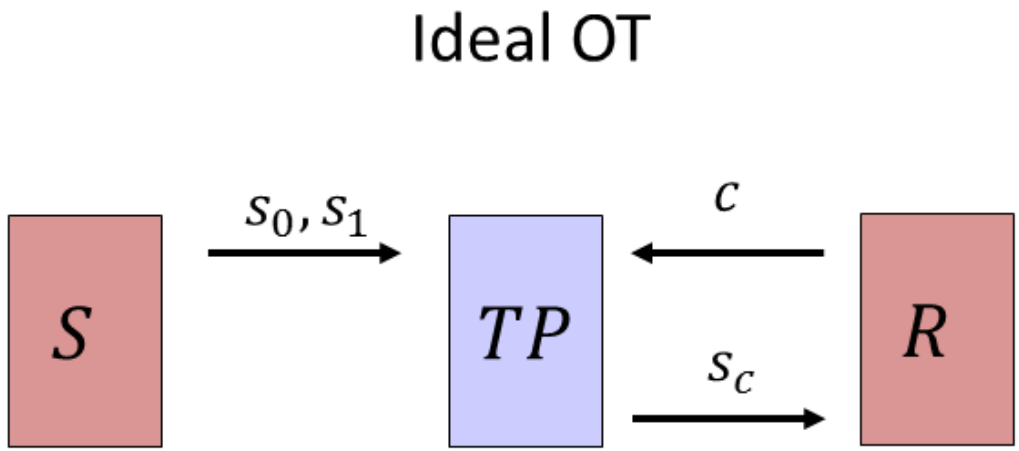
Secure Multi Party Computation: Privacy Requirements

- Computationally unbounded participants (output reveals nothing not implied by output and input).
- Computationally bounded participants (can simulate communication given the output, $f(x, y)$ and the input of one participant, x or y.

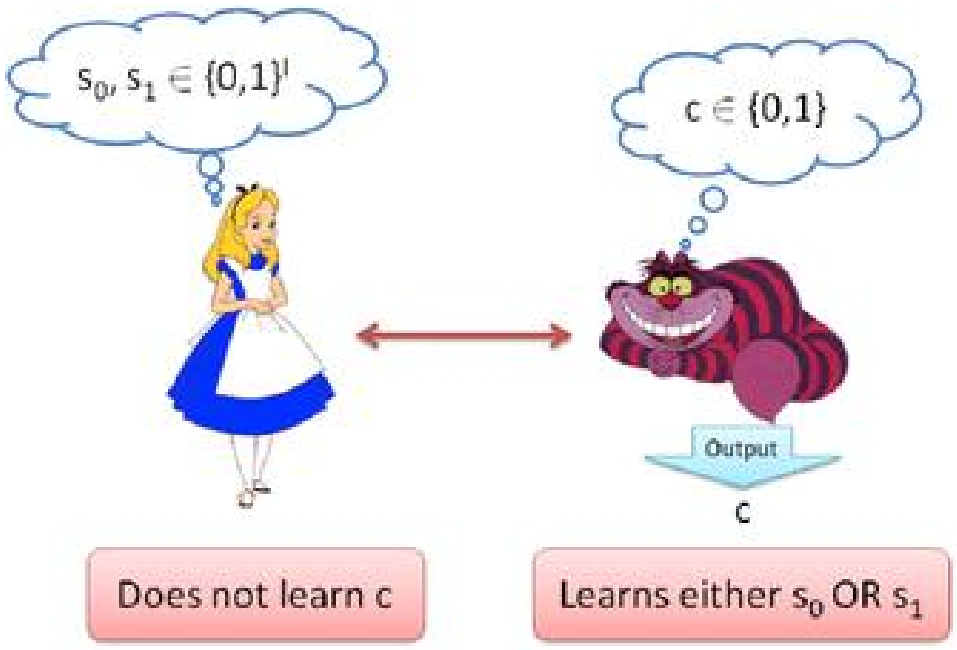
Computationally Unbounded Parties: The Millionaires' Problem

- Problem description.
- Impossibility of solution for computationally unbounded, honest but curious parties.
- A combinatorial characterization of 2 party privately computable functions.

Oblivious Transfer (OT): Ideal Setting (trusted third party) (up and coming, anonymous, etc.)



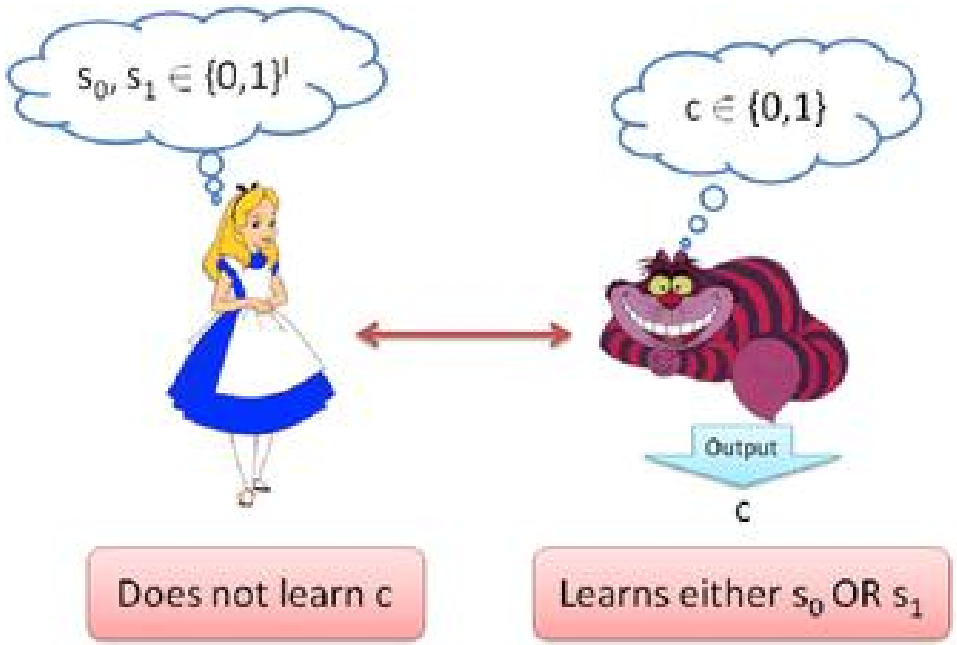
Oblivious Transfer (OT): Real Setting



(figure taken from this quantum computing site)

Oblivious Transfer (OT)

- A fundamental two party primitive.
- First version proposed by Michael Rabin, 1981.



- Most useful version: 1-out-of-2 OT.
- Can be based on either any trapdoor permutation, or specific assumptions such as decisional Diffie Hellman.

Representing Functions as Boolean Circuits

- Let $F : \{0,1\}^* \longrightarrow \{0,1\}$ be a Boolean function.
- Suppose it is computable by a TM in time $T(n)$.
- The "n-th slice" of F, *i.e.* $F : \{0,1\}^n \longrightarrow \{0,1\}$, can be computed by an $O(T^2(n))$ size Boolean circuit, C_n, with n inputs and Boolean gates of fan-in 2.
- The circuit C_n can be efficiently computed, given the description of the TM and 1^n, which is n in unary (think of the tableau in Cook's theorem, without the non-determinism).

Secure Function Evaluation: Computationally Bounded Case

- Problem description.
- Yao's garbled circuit.
- A solution for 2 honest but curious participants.
- Extensions to more participants and to malicious participants (brief sketches)

Yao's Garbled Circuit Evaluation: High Level View

- Let C_{2n} be a circuit computing $F : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}$.
- Two honest but curious parties, Alice and Bob, who know and agree upon C_{2n}.
- Alice's input are the n bits $x_1, x_2, \ldots, x_n$.
- Bob's input are the n bits $y_1, y_2, \ldots, y_n$.
- They wish to compute $F(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)$ correctly and privately.

- Alice constructs. Bob evaluates.

Yao's Garbled Circuit Evaluation: High Level View

- Alice constructs. Bob evaluates.
- For each wire w in the circuit, Alice produces two secret keys, $k_{0,w}$ and $k_{1,w}$. She keeps them, well, secret.
- These keys correspond to the value of the wire w being 0 or 1, correspondingly.
- The keys are chosen at random, and in particular they are independent of the input values to C_{2n}.

Yao's Garbled Circuit Evaluation: High Level View, cont.

- For each wire w in the circuit going out of a gate (namely all except the input wires), Alice produces four encryptions of the two keys $k_{0,w}$ and $k_{1,w}$.
- The four encryptions correspond to the four possible combinations of the two inputs to the gate that w goes out of.
- These encryptions are garbled, or permuted, such that Bob does not know what value (in $\{0,1\}$) they correspond to.
- Bob will be able to learn exactly one of the two keys $k_{0,w}$, $k_{1,w}$.
- For each input wires, Bob learns the key $k_{b,w}$ corresponding to his input bit b, but not the other key.
- How? Using 1-out-of-2 oblivious transfer with Alice.

Yao's Garbled Circuit Evaluation: A Separate Presentation

Maybe even two for the price of one

And Now to Something Completely Different: Secret Sharing and Error Correction

Three Basic Challenges in Communication

1. Secure (confidential) communication over insecure channels.

Three Basic Challenges in Communication

1. Secure (confidential) communication over insecure channels.

2. Reliable communication over unreliable (noisy) channels.

Three Basic Challenges in Communication

1. Secure (confidential) communication over insecure channels.

2. Reliable communication over unreliable (noisy) channels.

3. Frugal (economical) communication over expensive channels.

Three Basic Challenges in Communication

1. Secure (confidential) communication over insecure channels.

   Solved using cryptography (encryption/ decryption).

2. Reliable communication over unreliable (noisy) channels.

   Solved using error detection and correction codes.

3. Frugal (economical) communication over expensive channels.

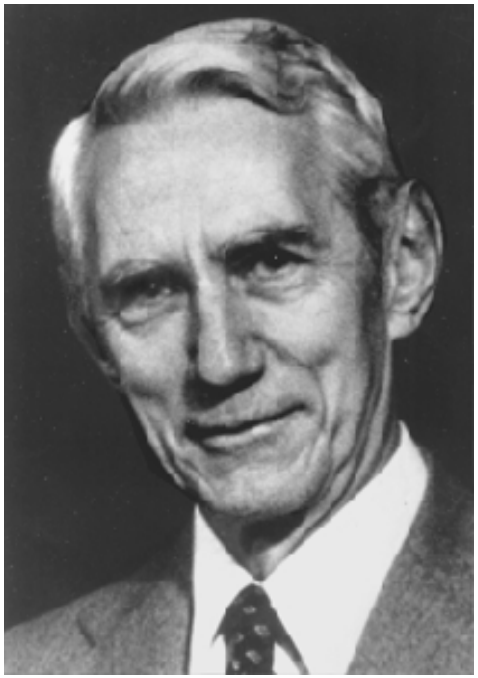
   Solved using compression (and decompression).

Each requirement can be treated separately. Of course, in a real scenario, solutions should be combined carefully so the three challenges are efficiently addressed (*e.g.* usually compression should be applied before encryption).

Today, we will very briefly discuss error detection and correction codes.

Claude Shannon, the Father of Information Theory

Claude Elwood Shannon (April 30, 1916–February 24, 2001) was an American mathematician, electronic engineer, and cryptographer known as "the father of information theory".



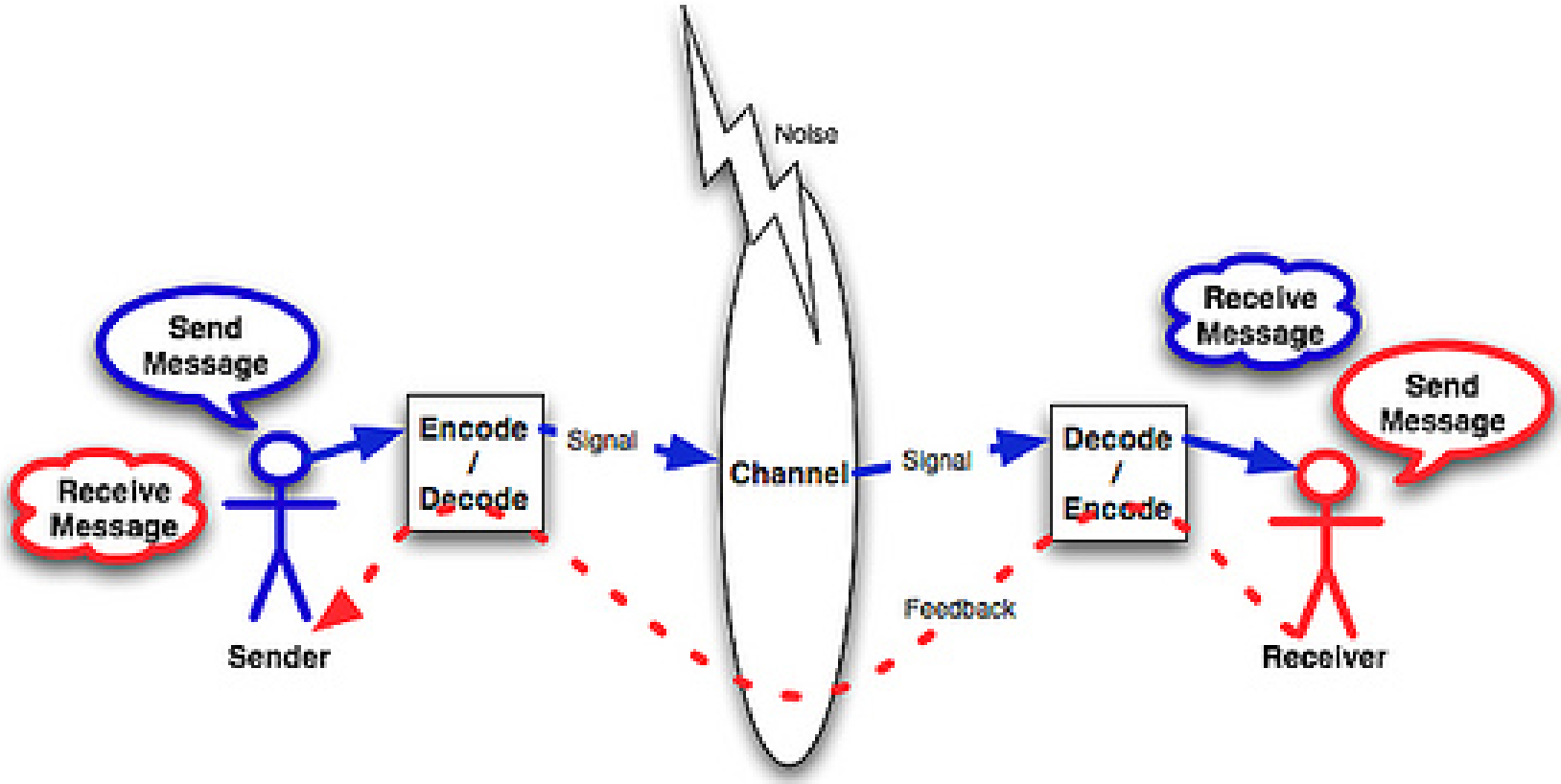
Shannon is famous for having founded information theory with one landmark paper published in 1948. But he is also credited with founding both digital computer and digital circuit design theory in 1937, when, as a 21 year old master's student at MIT, he wrote a thesis demonstrating that electrical application of Boolean algebra could construct and resolve any logical, numerical relationship.

It has been claimed that this was the most important master's thesis of all time. Shannon contributed to the field of cryptanalysis during World War II and afterwards, including basic work on code breaking.

For two months early in 1943, Shannon came into contact with the leading British cryptanalyst and mathematician Alan Turing. Turing had been posted to Washington to work with the US Navy's cryptanalytic service.

(text from Wikipedia)

The Shannon-Weaver Model of Communication (1949)

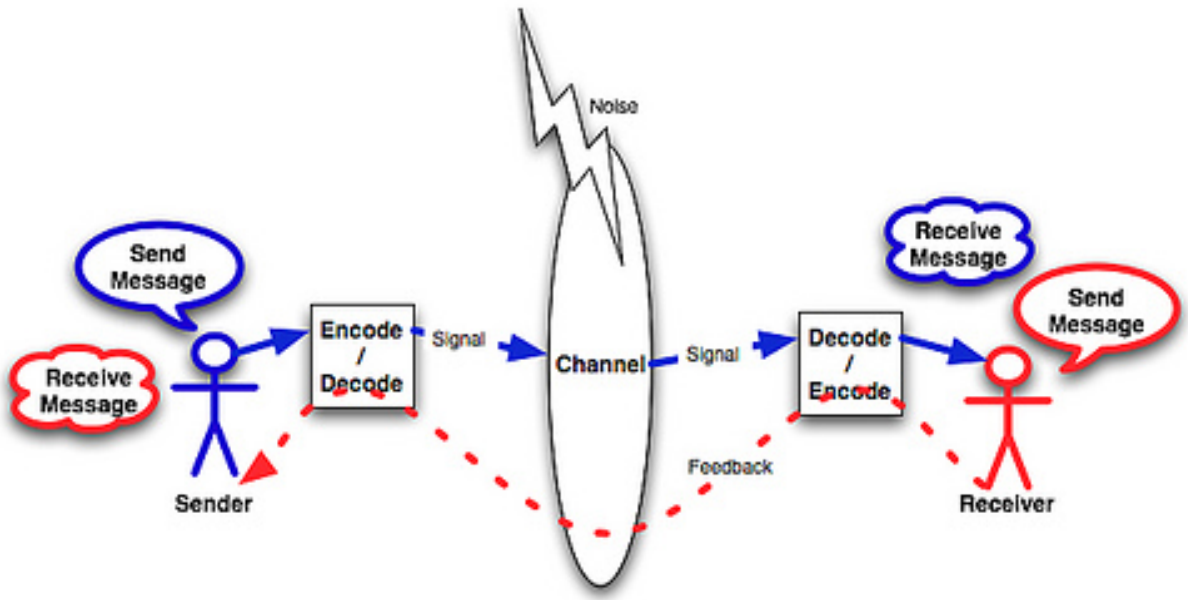


Source of figure is somewhat unexpected.

The Shannon-Weaver Model of Communication (1949)

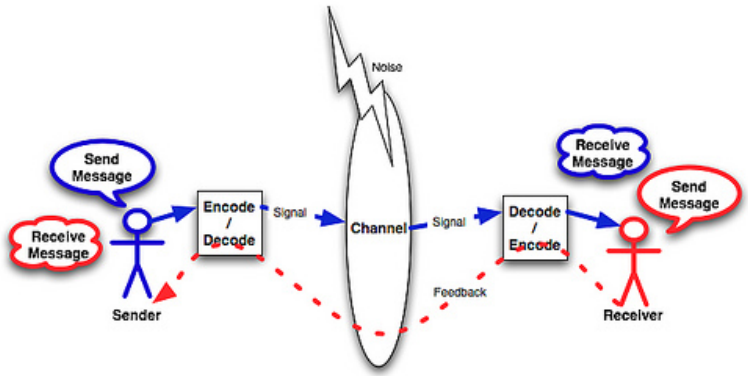
"We may have knowledge of the past but cannot control it;
We may control the future but cannot know it.".

Claude Shannon, 1959



For simplicity, let every original message be a fixed length block of characters from Σ. The channel is noisy, so a subset of sent bits may get altered (reversed) along the way, with non-zero probability.

The Shannon-Weaver Model of Communication, cont.



Sender passes original message through an encoder, which typically produces a longer signal by concatenating so called parity check bits (which may, of course, get altered themselves).

The (possibly altered) signal reaches the recipient' decoder, which translates it to a message, whose length equals the length of the original message.

Goal: Prob(original message equals decoded message) ≈ 1.

Hamming Distance

Richard W. Hamming (1915 –1998).

- Let $x, y \in \Sigma^n$ be two length n words over alphabet Σ. The Hamming distance between x, y is the number of coordinates where they differ.
- The Hamming distance satisfies the three usual requirements from a distance function
  1. For every x, $d(x, x) = 0$.
  2. For every x, y, $d(x, y) = d(y, x) \geq 0$, with equality iff $x = y$.
  3. For every x, y, z, $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality).

  where $x, y, z \in \{0, 1\}^n$ (same length).
- Examples
  1. $d(00101, 00101) = 0$
  2. $d(00101, 11010) = 5$ (maximum possible for length 5 vectors)
  3. $d(00101, 1101011)$ is undefined (unequal lengths).
  4. $d(\texttt{BEN}, \texttt{RAN}) = 2$

Definitions and Properties

An encoding, E, from k to n ($k < n$) bits, is a one-to-one mapping $E : \{0,1\}^k \mapsto \{0,1\}^n$. The set of codewords is the set $C = \{y \in \{0,1\}^n \mid \exists x \in \{0,1\}^k, E(x) = y\}$. The set C is often called the code.

Let $\Delta(y,z)$ denote the Hamming distance between y, z.
Let $y \in \{0,1\}^n$. The sphere of radius r around y is the set $B(y,r) = \{z \in \{0,1\}^n \mid \Delta(y,z) \leq r\}$.

The minimum distance of a code, C, is $\Delta(C) = \min_{y \neq z \in C} \{\Delta(y,z)\}$.

More Definitions (for reference only)

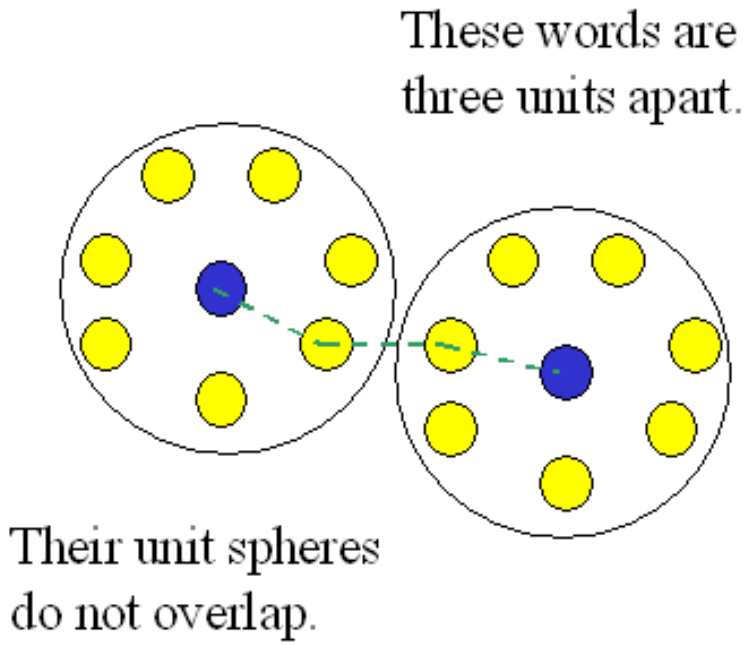
We say that a code, C, is capable of detecting r errors if for every $y \in C$, $B(y, r) \bigcap C = \{y\}$.

We say that a code, C, is capable of correcting r errors if for every $y \neq z \in C$, $B(y, r) \bigcap B(z, r) = \emptyset$.

(do the last two definitions make sense?).

An Important Observation

Proposition: Suppose $d = \Delta(C)$. Then the code, C, is capable of detecting up to $d - 1$ errors, and correcting up to $\lfloor (d-1)/2 \rfloor$ errors.



(figure from course EE 387, by John Gill, Stanford University, 2010.)

Secret Sharing and Error Correction

- Let $f(x), h(x)$ be two polynomials of degree up to $t-1$ over a finite field, F.
- We represent the polynomials by $(f(g), f(g^2), \ldots, f(g^n))$ and $(h(g), h(g^2), \ldots, h(g^n))$, both over F^n, where g is a multiplicative generator of F^*.
- The values at t points uniquely determine a polynomial of degree up to $t-1$.
- Hence, these two vectors could agree on at most $t-1$ entries.
- And their Hamming distance is at least $n-t+1$.
- Thus the code, C, is capable of detecting up to $n-t$ errors, and correcting up to $\lfloor (n-t)/2 \rfloor$ errors.
- In the context of error correction, this is known as the Reed–Solomon code.