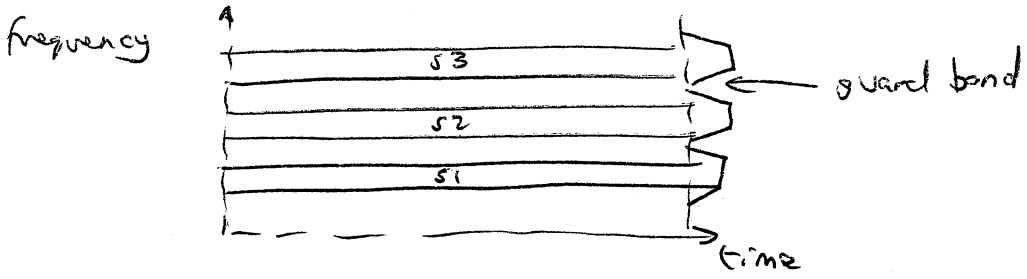
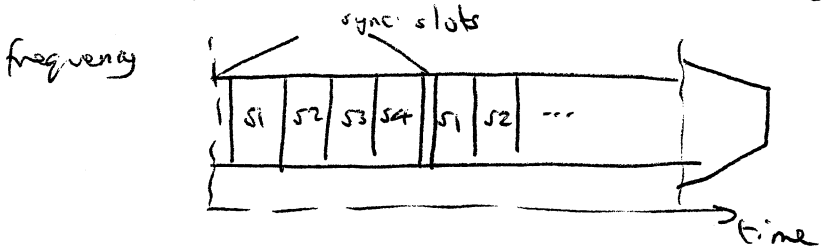
1 a) FDMA

Each signal is modulated on a separate carrier.



TDMA

Signals (usually digital) are interleaved in time and transmitted over a single serial link at a high symbol rate.

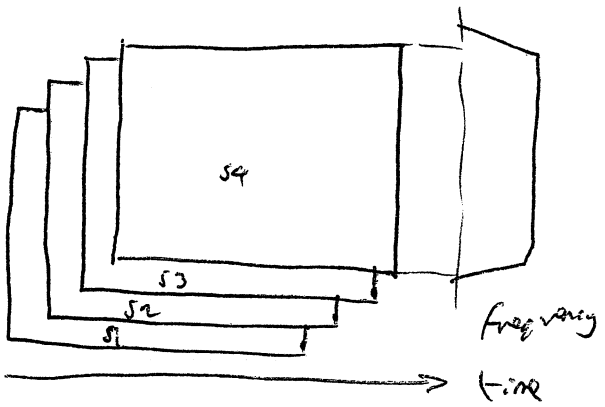


SSMA.

In SSMA each signal is further modulated by a spreading signal (also known as code modulation) to spread its energy over a much wider bandwidth. Many users may share the same frequency band provided different users employ separate codes with low cross-correlation properties.

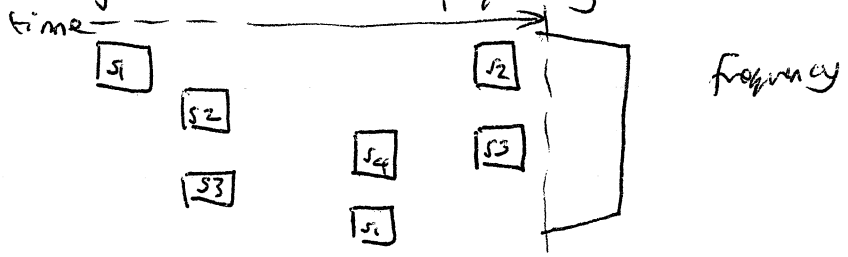
Direct Sequence

Signals overlap in frequency and time - achieved using a code with a high chip rate



Frequency Hopping

Signals use disjoint time and frequency cells.



FDMA

Advantages

- Simple to implement
- Little cooperation required between users.
- Good for broadcast, - cheap RX, separate TX to avoid intermod

Disadvantages

- Practical filters limit spectral efficiency
- Use of a common amp for multiple carriers gives rise to intermodulation interference
- For a BS, each channel requires its own mod/demod which is expensive

TDMA

Advantages

- Low implementation cost for digital signals - multiplexers are cheap
- High bandwidth efficiency since low overhead
- Can use non-linear (i.e. efficient) PA stages if const envelope modulation is used.
- Allow mobiles to TX and RX alternately - eliminating expensive duplex filter.

Disadvantages

- High level of coordination and control required since signals coming from TX at different distances must be timed to arrive within the defined rx window.
- High bit rates required - can give rise to ISI owing to multi-path radio channel. To overcome this problem hybrid FDMA/TDMA is used in GSM.

SSMA

Advantages

- Difficult to jam
- " " detect
- Requires little cooperation between groups of users
- Resistant to multipath (frequency selective) owing to wide bandwidth.

Disadvantages

- Can give less efficient use of spectrum
- FEC usually required to overcome high levels of interference from other users.
- Signal acquisition can be time consuming.
- Complex receivers
- Need separate bands to avoid interference from non-spread signals.

b) Channel bandwidth = 3.84 MHz.
So chip rate (assuming BPSK modulation) =
$R_c = \frac{3.84\ \text{Mchip/s}}{2} = 1.92\ \text{Mchip/s}$.

Now, $\frac{E_b}{N_0} = \frac{1}{n_u - 1} \cdot \frac{R_c}{R_b}$

$\therefore \quad R_b = \frac{1}{n_u - 1} \cdot R_c \cdot \frac{1}{\left(\frac{E_b}{N_0}\right)}$

$\left(\frac{E_b}{N_0}\right)_{dB} = 4.77 \quad \therefore \quad \frac{E_b}{N_0} = 3$

$\therefore$

$R_b = \frac{1}{43-1} \cdot 1.92 \times 10^6 \times \frac{1}{3}$

$= 15238\ \text{b/s}$.

This is a spreading ratio of $\frac{1.92 \times 10^6}{15238} = 126$

128 is the nearest available ratio (can't go lower than this otherwise will not be able to achieve the required no. of users.

So, the raw bit rate per user $= \frac{1.92 \times 10^6}{128}$
$= 15\text{kb/s}$.

With 1/2 rate coding, the user data rate is
$15\text{kb/s} \times 0.5 = \underline{\underline{7.5\text{kb/s}}}$

4f5 2003-2004
2 questions by Frank Stajano

Questions

Question 2

a. Imagine the many possible location-based services that could be offered to users of mobile phones. Describe the three you would consider most useful. [30%]

b. For each of the three services you presented in step (a), discuss whether it can be provided anonymously and to what extent. If so, how could this be done technically? What parts of the system would the user need to trust? (Solutions that minimize the Trusted Computing Base are obviously better.) [50%]

c. Mechanical locks that can be operated by different keys, with each key opening a particular subset of the deployed locks, are common in large buildings. Nowadays, though, this functionality can also be provided by electronic entry cards. Discuss the security aspects involved in the replacement of locks with entry cards, including location privacy and ways to preserve it. [20%]

Question 3

a. In the context of cryptographic primitives:
   i. What is a "blind signature"?
   ii. Describe a possible use for it.
   iii. Explain how it can be implemented. [40%]

b. We are interested in a "digital cash" protocol for anonymous payments over the Internet.
   i. What does such a protocol achieve? Who are the participants? In what way is the payment anonymous?
   ii. Describe the inner workings of such a protocol.
   iii. What are the security goals of the participants of the protocol? (Hint: they may be different for different participants.) How is your protocol protecting them?
   iv. Discuss the properties of your protocol compared with standard "Internet banking".
   v. Discuss the properties of your protocol compared with real (physical) cash. [60%]

4f5 2003-2004
2 questions by Frank Stajano

Cribs

Question 1

a. Very open question. Examples might include:

Dynamic on-demand mapping. Phone a shop and it will tell you or show you how to get there from where you are, sending you a graphical map plus textual instructions, hints about parking and so on.

"Where are the absent ones?" People agreeing to meet at a certain time form a "group" and everyone in the group can, at the time of the appointment, see at a glance where the latecomers are (instead of at the meeting place where they were supposed to be).

Alert me when I am near entities of a specific class. Whenever I am withing 50 m of a bookstore (or museum, or petrol station, or whatever), warn me, tell me what it's called and where exactly it is.

b. Can't give a pre-made solution as it depends on what was produced in (a). Anyway, valid points to be made include the following.

At some level, the location subsystem of the phone network operator knows the location of every phone. So it must be trusted by the user not to give it out to third parties.

It is easy to devise solutions in which all the location service is provided by the phone operator, who must know everything anyway. But this leads to a bloated Trusted Computing Base, so it is undesirable.

A better design is to farm out as much of the application-level location services to external untrusted applications, who receive appropriately obfuscated location information from the trusted operator.

It may be ok to give out an isolated location sighting if this is not accompanied by the identity of the phone holder.

However, depending on resolution, from a sequence of correlated location sightings it may be possible to guess the identity of the user.

c. A mechanical key is a token authorizing the holder to open a well-defined subset of doors. So is an entry card.

With the entry card system, it is easy to define new sets after the locks are in place. With the mechanical system this is either very hard or impossible without changing the locks.

With the entry card system it is possible to revoke a token without replacing the locks. This is impossible with the mechanical system.

A mechanical lock does not remember the keys that opened it. Holders of authorized tokens are let in and others are not, but there can be no logging. This is a guarantee of location privacy that doesn't exist with entry cards.

The most natural design for an entry card system is one in which a central system has a sparse matrix of all the issued keys and

all the deployed locks, saying what opens what. This design makes it easy to redefine sets and revoke keys as mentioned above. Every attempt to open a lock is a query into this central matrix, and can be logged.

It is of course possible to use an electronic system without any logging in order to protect location privacy, but it will be very hard to prove to a user that no logging is taking place.

The centralized design above has the disadvantage that locks stop working if the connection to the central system goes down. Therefore, for robustness, it may be preferable to split the central matrix into columns, one per lock, and download each column of allowed keys into the corresponding lock. At each attempt to open a door, the lock checks its local column vector and either opens or doesn't, but without talking to the central system. Logging, again, is optional (and invisible). Redefinition of sets and revocation of keys is still done on the central matrix, whose columns are then re-downloaded to the relevant locks after each change.

Question 2

a. See SCHNEIER, 5.3, for background.

i. A blind signature is a signature in which the signer does not see the document she is signing. Besides, even if the signer keeps a full track of all the signatures she makes, when she later sees the signed document she cannot recognize on what occasion she signed it, even though she is sure that she actually signed it.

ii. This can be useful when the signer needs to validate a credential but without learning its exact content.

Example: "Dear embassy, Mr Smith is one of our secret agents; please support him in his covert operations. Signed, the President." The President wants to be able to sign this letter without learning the fake cover name (Smith) that the agent chose for himself.

At the same time, the President wants to be sure that she is signing a letter of that format, not a forged one saying "Give this agent a seven-figure pension."

iii. One way to ensure the latter property is with a cut and choose protocol. The agent prepares 100 letters with different cover names; of these, the President checks 99 randomly chosen ones; if all ok, she signs the remaining one blindly.

The concept of blinding is to obfuscate the message before presenting it for signature, then remove the obfuscation from the signed message. To do this, one must exploit some tricky property of the particular signature scheme.

b. See SCHNEIER, 6.4, for background.

i. We have 3 participants: Customer, Bank, Merchant. The Customer has an account with the bank with some "real money" in it. The bank can issue the customer a bit string acting as anonymous digital money. The customer sends this string to the merchant who accepts it as payment in exchange for goods. The merchant then returns this string to the bank in exchange for "real money". The protocol is anonymous in the sense that the bank, on receiving the bit string from the merchant, can't tell who the customer was, even though it's the bank who gave the bit string to the customer in the first place. So the bank doesn't know what its customer is buying.

ii. Here is one possible protocol.

Customer prepares a message of this form "I, Bank, will pay 100 pounds to the first person who gives me this bit string, whose serial number is 1234231."

Customer blinds the bit string and submits it to Bank for blind signature, using cut and choose. So there will be many strings, all for the same amount but with different serial numbers.

Bank checks all but one of the strings, debits Customer's account by 100 pounds, signs blindly the remaining string (whose serial number it therefore doesn't know) and returns it to Customer.

Customer spends the bit string with Merchant. Before accepting it, Merchant phones Bank and asks whether Bank will honour it. Bank checks the signature (if it fails, Bank didn't sign it, so refuse) and checks that it hasn't already paid for that serial number. If all ok, it credits Merchant's account by 100 pounds, and Merchant proceeds with the sale. If problems, Bank refuses the bit string and so does Merchant.

iii. Customer wants to be sure that the bit string he gets from Bank will be accepted by Merchant. The protocol allows him to check Bank's signature when Bank issues the bit string. It also allows him to generate his own random serial number, to guarantee freshness (but note that, if Customer is unlucky enough to choose a serial number that was already claimed by someone else in the past, his bit string will be refused by the merchant.)

Customer also wants to be sure that if he pays Merchant, then Merchant will deliver the goods. The protocol does not guarantee this: Merchant could deposit the money and not deliver anything to Customer.

Merchant wants to be sure that Bank will accept the bit string. The protocol allows him to verify this online before proceeding.

Bank wants to be sure that what she blindly signs is a bit string for 100 pounds and not 10,000. The cut and choose portion of the blind signature sub-protocol ensures that.

Bank wants to be sure that a given bit string can only be cashed in once. The protocol ensures this through the serial number. However, if a particular bit string is presented twice, the bank cannot tell (for the purpose of prosecution) whether it was Merchant or Customer who attempted to double-spend.

Bank wants to be sure that Merchant or Customer won't make up random strings (since Bank doesn't know the serial numbers of the strings it issued until they are presented for payment). Bank's signature, which only Bank could have made, ensures this.

iv. With internet banking, Bank obviously knows the beneficiary of every payment made by Customer; so payments are not anonymous.

With our digital cash, the payment amount must be one of several "standard values" such as 100 pounds instead of 106.95--otherwise anonymity might be lost. So there is the additional problem of having to use many bit strings per payment (or the even more complex one of getting change).

v. Real cash can be accepted by Merchant without the need for an online transaction with the bank (i.e. the verification, with UV lamps or special pens, is local and offline).

Real cash can be transferred between two parties without having to

return it to Bank first.

With cash, Bank could keep track of the serial number whenever it issues a banknote to Customer (e.g. at every ATM withdrawal); and then correlate these serial numbers with the ones deposited by Merchants. With our digital cash this can't be done. However, as just noted, a merchant can spend real cash without first returning it to the bank, so if Bank gives me a particular banknote and then gets it back a while later from a fishmonger, it doesn't necessarily follow that I spent it on fish, so real cash is still anonymous despite the serial numbers.

If Bank and Merchants collude, though (e.g. if all merchants also check the serial numbers of real cash at point of sale), real cash would no longer be anonymous. What about our digital cash? If we assume that Customer can buy from Merchant without revealing his identity (and the difficulty here is in doing so while still receiving the goods) then Customer remains anonymous even if Bank and Merchant collude; otherwise not.

As already noted above, our digital cash can only come in one of several standard denominations. However the same holds for real cash, so no advantages or disadvantages here.

A. Hopper.

Q.4.

Qn 4F5 Digital Communications 2004

Sensor-driven or "Sentient" computing makes possible many new applications of computer and communication systems. It is potentially useful for in-building environments.

(a) Outline the major hardware and software components used to implement an in-building "Sentient Computing System" which uses location information.
[25%]

(b) Give a software architecture and describe the most important components of the middleware.
[25%]

(c) What is a spatial monitor and how can it be designed to handle a very high rate of indexing events.
[25%]

(d) Give three potential applications of in-building location-aware systems.
[25%]

Crib

(a)
In-building location sensors
World (data) model that represents the state of the environment
Persistent distributed object system
Telemetry and resource monitors to all digital devices (computers, phones, PABXs etc)
Spatial monitoring service
Timeline-based data storage
Applications

(b)
Software counterparts of real-world entities implemented as persistent distributed objects using CORBA and Oracle RDBMS
Proxy server to deal with the speed mismatch between sensor data and database
Proxy server to deal with the potentially large number of objects in use
Spatial monitor incorporating the spatial indexer
Core services dealing with sessions, events, synchronisation
Data services relating to resources, people, building

(c)
Define 2-D regions around objects
Convert absolute location events to relative geometric events
Generate positive and negative containment and overlapping events
Use quad tree representation of contours
Only index to precision required by application

Distribute processing on multiple machines

(d)
Follow-me desktops (VNC)
Telephone rerouting
Virtual buttons
Augmented reality
Time-based data storage and indexing