Lecture 10: Quantum Crytography

Dr Iain Styles: I.B.Styles@cs.bham.ac.uk

Introduction

Last lecture, we:

- showed how entanglement allows us to “teleport” quantum states

In this lecture we will:

- show how quantum mechanics could lead to more secure cryptosystems

Classical Cryptography

- Most modern cryptosystems are based around the idea of a “key”
- The purpose of the key is to unlock the encoded message
- In private key cryptosystems, Alice and Bob agree on a key known only to them before sending the message
- In public key crytosystems, Alice encodes the message using Bob’s public key. Bob can then decode it using his private key
- Both systems have advantages and disadvantages
- Private key crytosystems are provably secure **if** the key is guaranteed to be secret, and is never re-used
- Public key cryptosystems rely on assumptions about the computational hardness of certain problems
- Quantum computation, if eventually feasible on a large scale would destroy most public key systems
- But it could lead to provably secure private keys

Private Key Cryptography

- Private key cryptography is the oldest and best known type of cryptosystem and was by far the most common until public key systems were invented in the 1970’s
- It’s very easy to understand: we will use the example of the **Vernam cypher**, which is typical of the genre
- Alice wishes to send an n-character message to Bob without anyone being able to intercept and read the message
- Before the message is sent, Alice and Bob agree on an n-character **private key**
- The private key is basically a random string of characters
- Alice encodes the message by simply “adding” the key string to the message string modulo the number of possible characters
- Alice then sends the message to Bob over a public channel
- When Bob receives the message, he simply subtracts the key from the encoded mesage to get the original message string:

A Simple Example

Alice wishes to send	B	I	R	M	I	N	G	H	A	M
	+	+	+	+	+	+	+	+	+	+
She adds the key	G	P	A	D	F	C	U	R	K	T
	=	=	=	=	=	=	=	=	=	=
The encoded message is	I	Y	S	Q	O	Q	B	Z	L	G
This is sent to Bob	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Bob receives	I	Y	S	Q	O	Q	B	Z	L	G
	−	−	−	−	−	−	−	−	−	−
Subtracts the key	G	P	A	D	F	C	U	R	K	T
	=	=	=	=	=	=	=	=	=	=
Giving the message	B	I	R	M	I	N	G	H	A	M

Private Key Cryptography

- There are some inherent problems with this system
- The key must be at least as long as the message, making it impractical for long messages
- A new key must be used each time, or it could be deduced using statistical methods
- Anyone can intercept the message, but they need the private key to decode it
- If the key is guaranteed secret, this is OK, and such systems are provably secure
- But it is very hard to distribute keys in a secure way
- This is compounded by the need to use a new key each time
- If you can send the key securely, then you could use the same channel to send the message!

Public Key Cryptography

- Public key cryptosystems have become increasingly popular since their invention in the 1970s for a very simple reason:
- You only need to generate one pair of keys, and one half of the pair is **public**
- Anyone who can find your public key can send you an encrypted message
- The other half of the key pair is **private** and is known only to the owner
- The private key is used to decode the message
- So if Alice wishes to send Bob an encrypted message, she encodes it using Bob's public key
- The encrypted message is sent to Bob, who decodes it with his private key
- To decode the message, an eavesdropper must know the private key - the public key cannot be used directly to decode the message

RSA Cryptography

- RSA is the most common form of public key cryptography
- It is based on the properties of large numbers and works as follows:
- Bob picks two large prime numbers p and q, and uses them to compute $n = p \times q$
- He selects at random a small odd integer, e which is relatively prime to $(p-1)(q-1)$ — this means that e and $(p-1)(q-1)$ have no common factors except 1
- He then computes d such that $ed = 1 \mod (p-1)(q-1)$ (there are known algorithms to find d)
- The pair (e, n) form the public key
- The pair (d, n) form the private key

RSA Cryptography

- The key to RSA is that Alice needs only to know the public key to send Bob an encrypted message
- If Bob has a brain, he keeps his private key to himself so no-one else can read messages encrypted with his public key
- The decryption process relies on knowing both d and n.
- Clearly n can be found from the public key (e, n)
- In order to find d, you need to know the prime factors of n: p and q
- The security of RSA relies on our inability to efficiently factor large numbers: the number field sieve is exponential in n
- But Shor's algorithm on a quantum computer is *much* better!
- Shor's algorithm, if successfully implemented on a large scale, would spell the end of RSA and many other common cryptosystems

Quantum Key Distribution

- No radically new encryption schemes emerge from our studies of quantum information
- But QM can make our existing methods provably secure, in particular:
- We can use qubits to securely distribute private keys
- Quantum Key Distribution is provably secure
- There are several protocols: BB84, B92, EPR
- We will focus on BB84, due to Bennett and Brassard (in 1984!)

The BB84 Protocol

This is perhaps the simplest of QKD protocols

- Alice generates a random bit string b (could be classical, must be private). Say, $b = 10110101$
- Alice then forms the equivalent quantum string $|10110101\rangle$ and "encodes" it by passing each bit through either an identity or a Hadamard gate, chosen at random
- Say Alice chooses $HIHIHIHI$
- This gives the quantum state $|q\rangle = |-0 - 1 + 1 + 1\rangle$
- Alice then sends this state to Bob—not by teleportation, but using polarised photons down a fibre
- When Bob receives the state he randomly chooses whether to pass each qubit through an identity or a Hadamard gate, say $HHIIHHII$
- Bob then measures the state—where he chose the same "encoding" as Alice, the result of his measurement is certain
- In this case, Bob will measure $1??10??1$

The BB84 Protocol

- Alice and Bob then publicly announce the *encodings* they chose: Alice: HIHIHIHI; Bob: HHIIHHII
- They retain only those qubits for which they chose the same encoding, in this case, the first, fourth, fifth and eighth bits
- Alice publicly announces the values of a subset of the retained bits: e.g. bits 4 and 5: 10
- Bob checks to see whether his measurements were the same
- If so, they use the undisclosed bits as a private key for message exchange
- If not, they abort the communication
- How does this help us detect an eavesdropper?
- The key is that Eve must guess the encodings that Alice and Bob chose
- An incorrect choice will lead to Alice and Bob's checkbits failing to match

Detecting Eve

- Say Alice transmits $|q'\rangle = |-0-1+1+1\rangle$, encoded using $HIHIHIHI$
- Eve cannot know Alices encoding and must guess - say $HHHHIIII$
- Eve "decodes the intercepted signal", getting $|1+1-+1+1\rangle$ and then measures
- Eve will measure $1?1??1?1$—say 10100111
- Eve then reencodes with $HHHHIIII$, getting $|q'\rangle = -+-+0111$
- Where Eve's encoding matches Alice's, the measurement does not change the state and Eve's re-encoding reproduces $|q\rangle$
- When Eve's and Alice's encodings do not match, the quantum string that Bob receives will not be the one Alice sent
- Bob receives $|q'\rangle$, and decodes using $HHIIHHII$
- Bob measures $10????11$, for example 10111011
- Alice and Bob then publicly compare bits 4 and 5— Alice: 10; Bob: 11
- Eve's interception has corrupted the key, and Alice and Bob know it has not been transmitted securely

Some Notes

- BB84 is not an encryption protocol
- It is simply a key exchange mechanism
- BB84 allows a private symmetric key to be generated probabilistically
- The actions of both Alice and Bob determine the key: it is not transmitted from one to the other
- Obviously this exchange protocol cannot be used to exchange a message directly as the key is random
- BB84 could be used to generate keys for any symmetric-key encryption protocol
- Allows a unique key to be generated for each message exchanged
- It can be shown that if unique keys are used, then even very simple ciphers are provably secure

Some Notes

- In the short example given, there is clearly a finite probability of Eve remaining undetected
- For a long-enough check-string, this probability is negligible
- The same ideas can also be used to detect noise
- In practice, an acceptable error rate is calculated and so long as the number of checkbit errors is small enough, the tranmission is validated
- This is feasible with current technology
- Longest distance in a fibre: 148.7km at LANL in 2007
- Longest distance in free space: 144km
- DARPA Quantum Network: 10 interconnected QKD hubs
- First computer network using QKD: six locations around Vienna - October 2008
- Transmission rates are quite low, but it is extremely reliable
- Noise levels are low enough to guarantee privacy

Conclusions

In this lecture we have:

- discussed different types of classical cryptosystem
- shown how qubits can be used to securely distribute private encryption keys

Next lecture we will:

- outline various ways in which one could construct a quantum computer