- Tasks that participants demand guarantees with respect to origin, privacy, integrity etc of a message or a computation

Examples:

1. **Encryption**: Two parties want to communicate with the guarantee that nobody else obtains any information about their communication.

2. **Authentication**: Two parties need to communicate with the guarantee that they “talk” to each other and that nobody has tampered their communication. The messages need *not* be secret.

3. **Digital Signature**: Guarantee the origin, authenticity and transferability of a message.

4. **Bit commitment**: Alice commits to a value of a bit without revealing her choice, but with guarantee that she cannot change her choice after the commitment.
Types of security

1. Computational Security
   - Relies on assumptions (possibly unproven) about difficulty of performing certain problems efficiently.
   - Security breaks if better (classical) algorithms are found, new devices (quantum computer), faster (classical) computers are constructed or given sufficient time.
   - Security can break retrospectively (revealing past secrets)

2. Information Theoretic Security ("unconditionally secure")
   - Relies on limitations imposed by nature, whether those are due to information theoretic constraints or the laws of quantum mechanics
   - Authentication of classical messages can be done with information-theoretic security using classical algorithms ("almost", need small shared seed). We can assume for remaining that parties share authenticated classical channels
Classical Encryption

- Alice wants to send a message $x$ of $m$ bits to Bob ("plaintext"), in such a way that nobody else (e.g. Eve - eavesdropper) learns anything about the message
- It can be proven that the only information theoretic secure way to encrypt a message is the one-time-pad

If Alice and Bob share a secret key $k$ of length $m$, then Alice encrypts the message adding (modulo 2) the key bit-by-bit to the message. This is the "ciphertext"

Alice sends the ciphertext through an untrusted channel to Bob

Bob decrypts the message by adding the key

Example:
Message $x = 101$, shared key $k = 011$

Alice sends $(1 \oplus 0)(0 \oplus 1)(1 \oplus 1) = 110$ is the ciphertext

Bob decrypts $(1 \oplus 0)(1 \oplus 1)(0 \oplus 1) = 101$
Classical Encryption

- The secret key cannot be re-used for encrypting other messages (thus “one-time-pad”)
- Need shared secret key between Alice and Bob (as long as the message) such that Eve has no information. This is known as Key Distribution
- Classically this can be done only with computational security. Existing KD implementations use the assumption that factoring or discrete logarithm is hard.
As we saw in last lecture, quantum computers can break this using Shor’s algorithm!!
- However, we can use quantum mechanics again to counter the threat of quantum computers!
Using QM we can distribute a key with information theoretic security based on the laws of QM: Quantum Key Distribution (QKD)
- Charles Bennett and Gilles Brassard in 1984 gave the first QKD protocol (photo: Bennett)

\[
|\pm\rangle = \frac{1}{\sqrt{2}} (|H\rangle \pm |V\rangle)
\]

- Followed “quantum money” paper (Stephen Wiesner)
- Set of states \( \{ |H\rangle, |+\rangle, |V\rangle, |-\rangle \} \)
- Alice encodes the classical bit 0 with either \( |H\rangle \) or \( |+\rangle \) and the classical bit 1 with either \( |V\rangle \) or \( |-\rangle \)
- Alice sends a string of qubits corresponding to an equal length string of classical bits
  E.g. $|H\rangle |+\rangle |−\rangle |H\rangle |V\rangle$ corresponding to the string 00101
- Bob randomly chooses either the basis $x_0 = \{|H\rangle, |V\rangle\}$ or the basis $x_1 = \{|+\rangle, |−\rangle\}$ and makes a measurement. He records the (classical) outcome $\{0, 1\}$ and the basis $\{x_0, x_1\}$ he measured
- Bob announces the basis $x_i$ he measured (NOT the outcome)
- Alice responds for which qubits Bob measured at the correct basis
- Alice and Bob discard all the qubits that Bob measured in the wrong basis
- For the remaining qubits, if no eavesdropping has occurred and no noise (ideal case), Alice and Bob should agree 100%. This is called the raw key
- They choose a small fraction (e.g. 10%) of the bits and check if they have any disagreements.

This phase, where Alice and Bob find what fraction of mismatches they have, is called **Parameter Estimation (PE)** phase.

- The qubits that are used in the PE phase are also discarded ("sacrifice" some part of the string to establish what correlation Alice has with Bob).

- The remaining bits (after discarding positions that Bob measured at the wrong basis and those that were used at the PE phase) will be used for distilling a secret key.

- In the ideal case (no noise), the latter is exactly the secret key that Alice and Bob share.
Example:

<table>
<thead>
<tr>
<th>Key Obtained</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding (Alice)</td>
<td>$</td>
<td>H\rangle$</td>
<td>$</td>
<td>+\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>Measurement (Bob)</td>
<td>$x_0$</td>
<td>$x_0X$</td>
<td>$x_1$</td>
<td>$x_1X$</td>
<td>$x_0X$</td>
</tr>
<tr>
<td>Outcome</td>
<td>$</td>
<td>H\rangle$</td>
<td>$</td>
<td>V\rangle$</td>
<td>$</td>
</tr>
<tr>
<td>Key Obtained</td>
<td>0</td>
<td>1$X$</td>
<td>1</td>
<td>1$X$</td>
<td>1$X$</td>
</tr>
</tbody>
</table>

- Neither Alice nor Bob can decide what key they will share by the end. Therefore QKD *generates* a secret key rather than distributes.
Forging attempts

- Basic principle behind this is the “no-cloning”. Eve cannot copy the states that Alice sends.
- Whenever she tries to distinguish between the BB84 states, she sometimes fails.
- Any measurement she does changes the quantum state and thus introduces errors that are detected in the PE phase.
- If Eve knew the basis that the state Alice sends belongs, she could forge. She would measure in the correct basis causing no disturbance, get always correct answer and then send to Bob the correct state.
- Eve could also forge if she could copy the unknown states send.
- However, Alice announces the basis AFTER Bob has received and measured his states.
- The full proof against any attacks is involved. Below we give a particular attack by Eve and how this introduces errors.
Forging attempts

**Attack:** Eve intercepts the quantum channel and measures the qubits randomly at the $x_0$ or $x_1$ basis. Then she prepares qubits according to the outcomes she obtain, and forward these to Bob. Then Bob, does not know that an adversary has acted and continues the protocol as he should.

**Effect:**
1. Alice sends a qubit, say $|H\rangle$
2. Eve intercepts and measures at the wrong basis $x_1$ with probability $p_1 = 1/2$.
3. Eve gets either $|+\rangle$ or $|-\rangle$. In both cases Bob will have the same probability to detect the attack.
4. Bob measures at the basis $x_0$. If he measures at the $x_1$ this qubit is discarded and does not “reach” the PE phase.
5. With probability $p_2 = 1/2$ Bob does not obtain the outcome $|H\rangle$ (independent of what Eve got at step 3).
6. The probability that an error is detected at the PE phase, for this attack is $p_1 \times p_2 = 1/4$.
- If the fraction of errors at the PE phase is much lower than $1/4$ we can say with certainty that the errors are not caused by the above attack
- There exist attacks that are specific to the physical implementation of the above (Photon Number Splitting, Trojan-horse attack, etc)
- In realistic implementations there are always few errors at the PE phase, due to noise
- However if the ratio of errors is small enough (e.g. around 10% and certainly less than $1/4$), we can still distil a secret key
- By the end of the PE phase Alice has a bit string $X$ and Bob another $Y$, while Eve has possibly a string $W$
- $X$ and $Y$ are correlated. If they are more correlated than either of these with $W$, then one can still achieve a common and secret key between Alice and Bob, following the following two extra steps. How correlated the strings $X$, $Y$, $W$ are, can be computed using the percentage of errors in PE and information theoretic measures
1. Information Reconciliation (IR): Exchange information (error-correcting codes) to make $X' = Y'$, while in the process some extra information is leaked to Eve

2. Privacy Amplification (PA): Using (a family of) universal hash functions make sure that the final (smaller) key that Alice and Bob share is completely secret from Eve (i.e. amplify the privacy). This is achieved since for hash function $h$, $h(X') = h(Y')$, but $h(X')$ is very different than $h(W')$.

- There exist many other QKD protocols. Entanglement Based protocols, helped prove formally the security of QKD against all attacks and lead to Device Independent QKD (next lecture)