The Post-Processing of Quantum Key Distribution

Hao Chung (鍾豪)
National Taiwan University

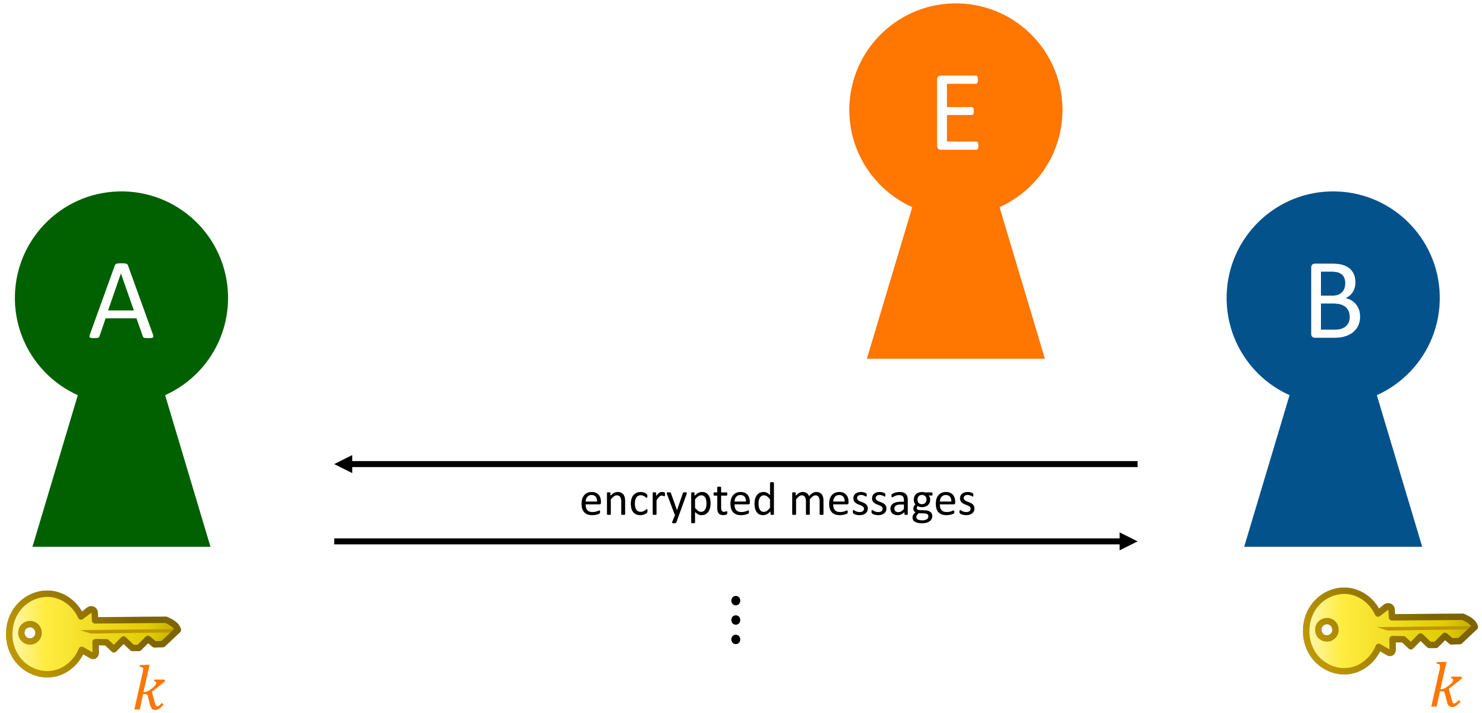
May 1, 2018

Outline

1. Quantum Key Distribution
2. Information Reconciliation (Error Correction)
3. Privacy Amplification (Randomness Extractor)

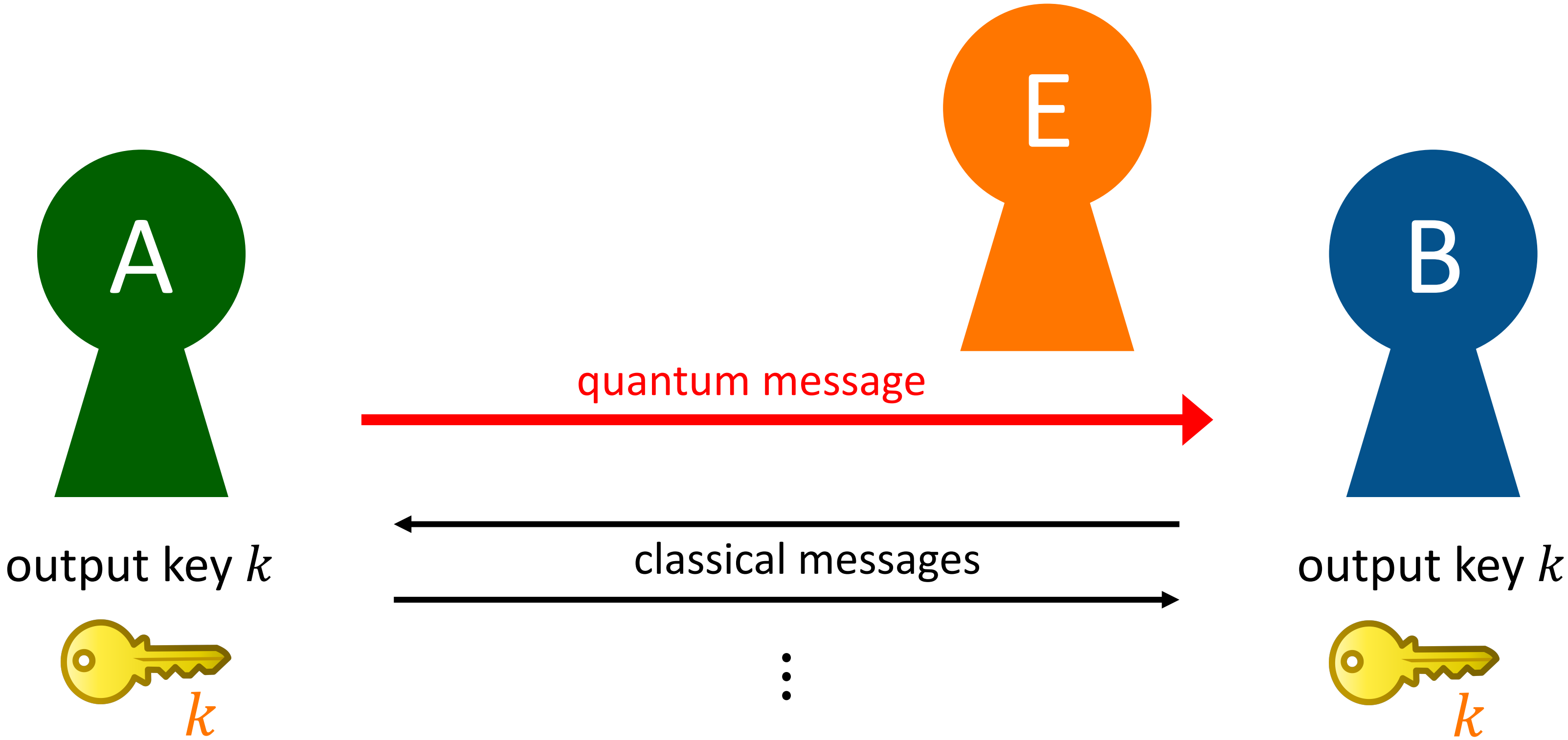
Key Distribution

To enable efficient secure encrypted communication,
Alice & Bob need to share a uniform key k against adversary Eve.
How do they establish such a shared key k?



Quantum Key Distribution

Allow Alice and Bob have a quantum channel.



Assume they have authenticated classical channel

Main Structure of QKD protocol

Encoding

Alice encodes information in some quantum signals and send them to Bob.

Parameter Estimation

Alice and Bob do measurements on quantum signals and discuss over the classical channel in order to estimate the error rate.

Information Reconciliation and Privacy Amplification

Alice and Bob apply some algorithm depending on error rate so that they can have a shared secret key.

QKD Setup

Alice and Bob can use two channels

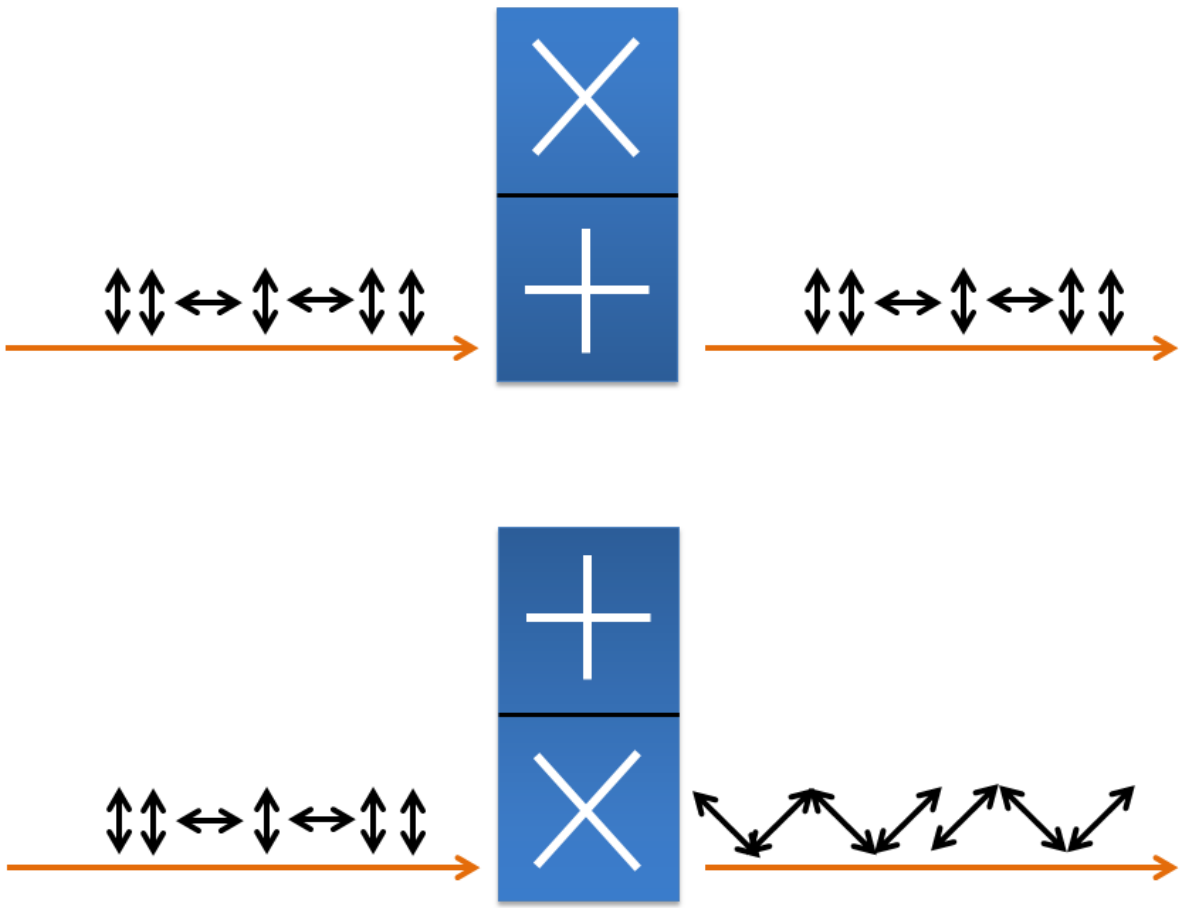
Channel	Bit 0	Bit 1
$\oplus$	$0^\circ(\uparrow)$	$90^\circ(\rightarrow)$
$\otimes$	$45^\circ(\nearrow)$	$135^\circ(\nwarrow)$

If a $|\uparrow\rangle$ is measured under $\oplus$ basis, the result will be $|\uparrow\rangle$ with probability 100%.

- The same goes for $|\rightarrow\rangle$ under $\oplus$ and $|\nearrow\rangle$ or $|\nwarrow\rangle$ under $\otimes$.

If a $|\nearrow\rangle$ is measured under $\oplus$ basis, the result will be $|\uparrow\rangle$ with probability 50% or $|\rightarrow\rangle$ with probability 50%.

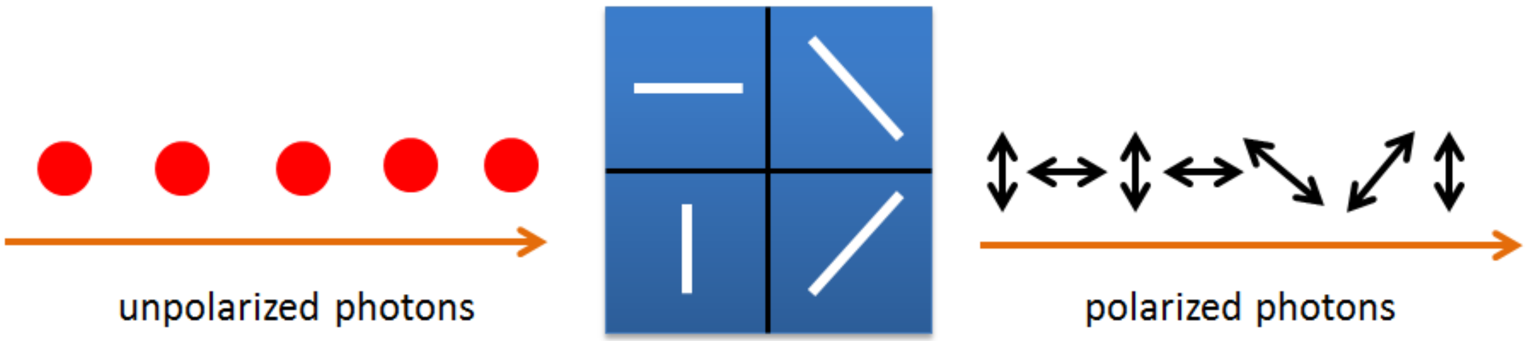
Encoding of BB84



Encoding of BB84

Alice sends polarized photons. Each photon polarizes at one of the four possibilities randomly.

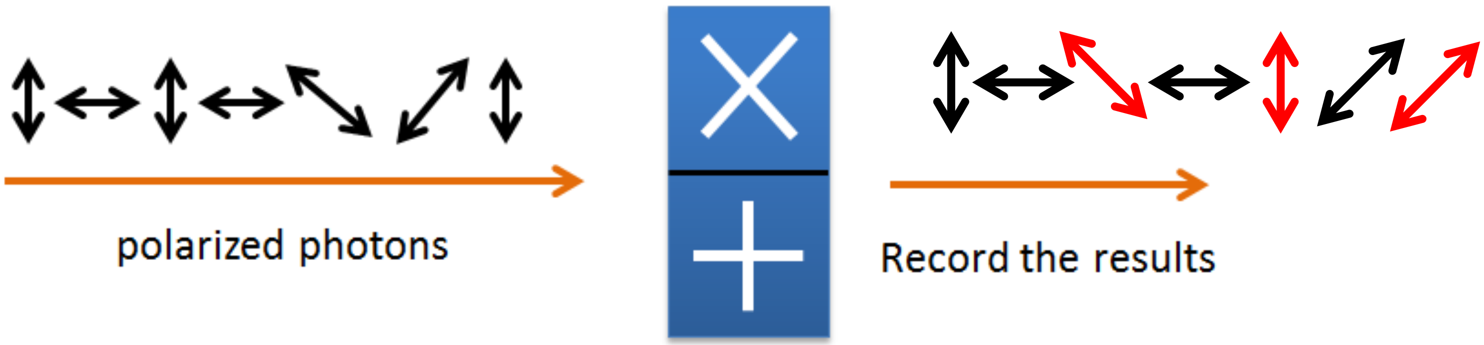
Alice doesn't tell anyone including Bob what basis that she chooses.



Parameter Estimation of BB84

Bob measures the photons using a random choice of two bases and records the results.

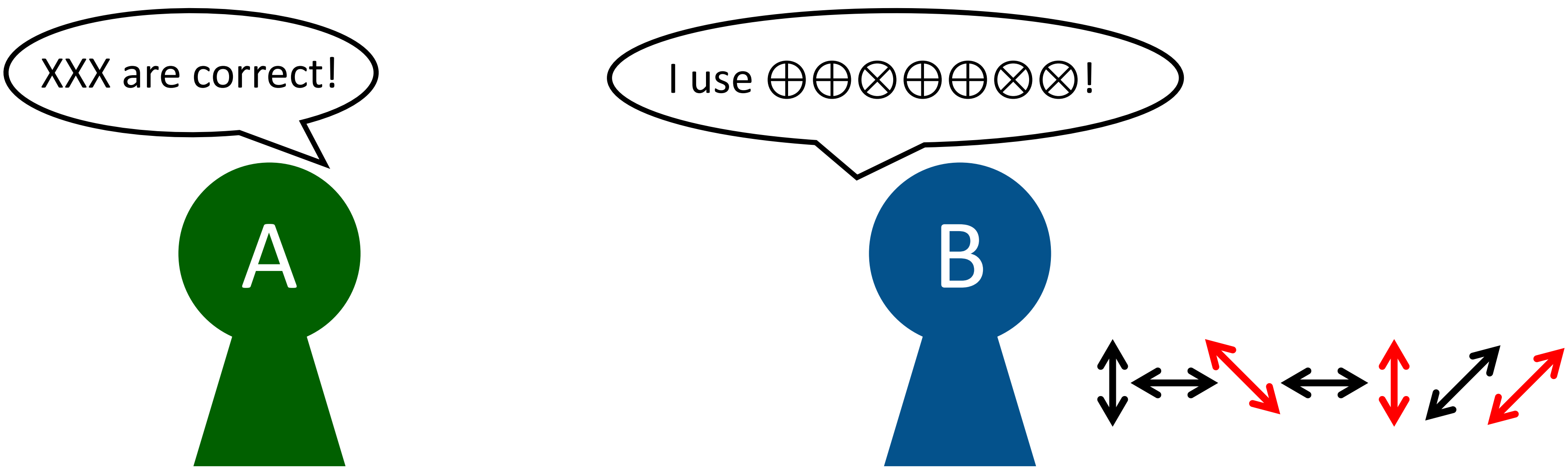
In average, half of the photons will be measured by wrong basis.



Parameter Estimation of BB84

Bob tells Alice which basis he applied for each photons in public channel.

Alice tells Bob which photons are measured correctly. Those photons are called “sifted photons” and other photons are aborted.



Parameter Estimation of BB84

Bob transmits some of the "sifted photons" to Alice.

Alice does the error analysis:

- If the channel is reliable, all the measured results should be the same.
- If the channel is eavesdropped, some results are inconsistent.

The sifted photons that are not used for error analysis are the raw key.

Example

A' data	1	0	0	1	1	1	0	0	1	0	0	1
A' basis	$\oplus$	$\otimes$	$\oplus$	$\otimes$	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\otimes$	$\otimes$	$\oplus$
θ	90	45	0	135	90	90	0	45	90	45	45	90
B' basis	$\otimes$	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\otimes$
B' result	1	0	0	0	1	1	0	1	1	0	1	1
Same basis?	n	y	y	n	y	y	n	n	y	y	n	n
Sifted bit		0	0		1	1			1	0		
Data check?		y	n		y	n			y	n		
Private key			0			1				0		

Reproduce from Wikipedia

Parameter Estimation of BB84

However, in practice, no error is almost impossible.

- Among the sifted photons, they choose a subset of the photons and compare the measurement results. If more than δ portion are different, they abort the protocol.

The goal of parameter estimation is to make a good upper bound of the error rate in the remaining photons.

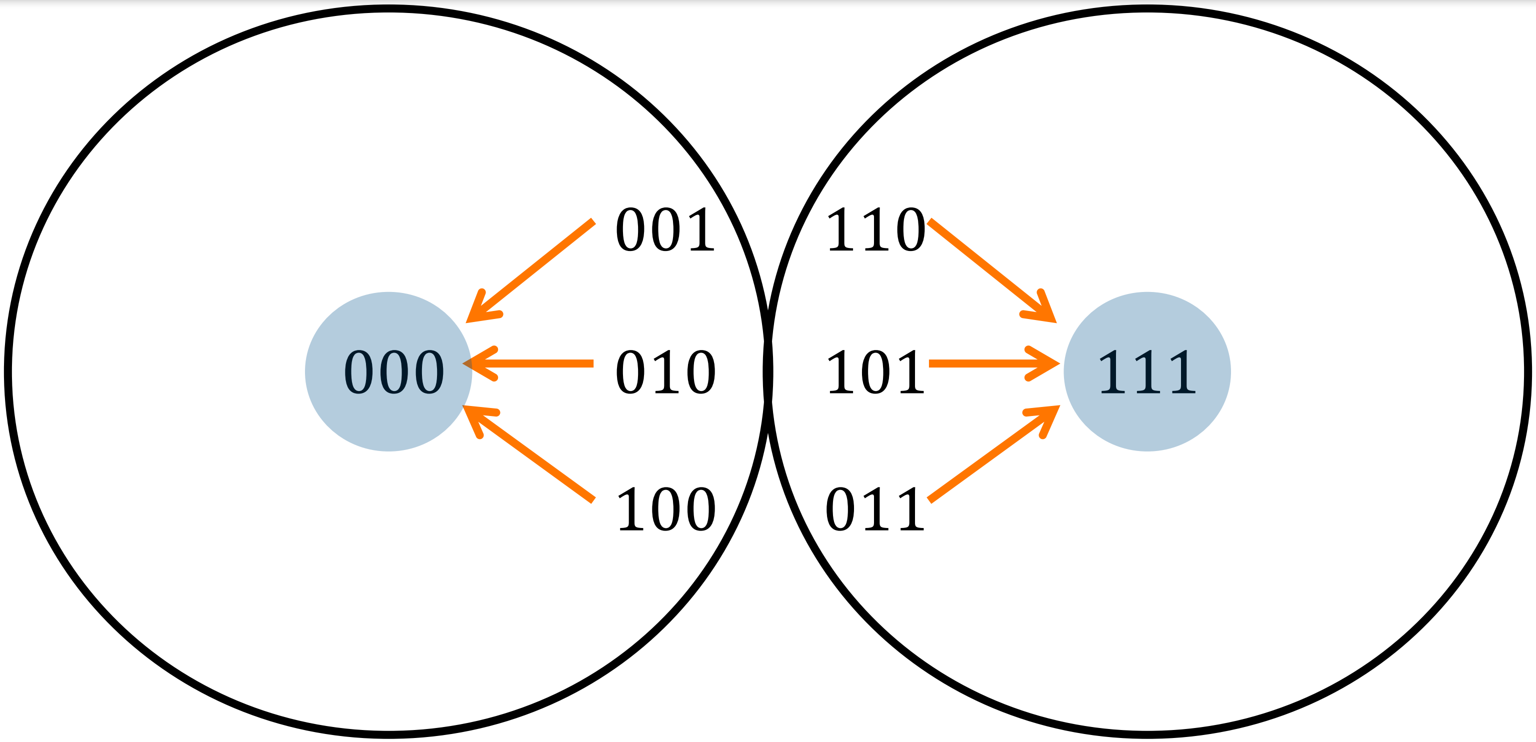
Outline

1. Quantum Key Distribution
2. Information Reconciliation
3. Privacy Amplification

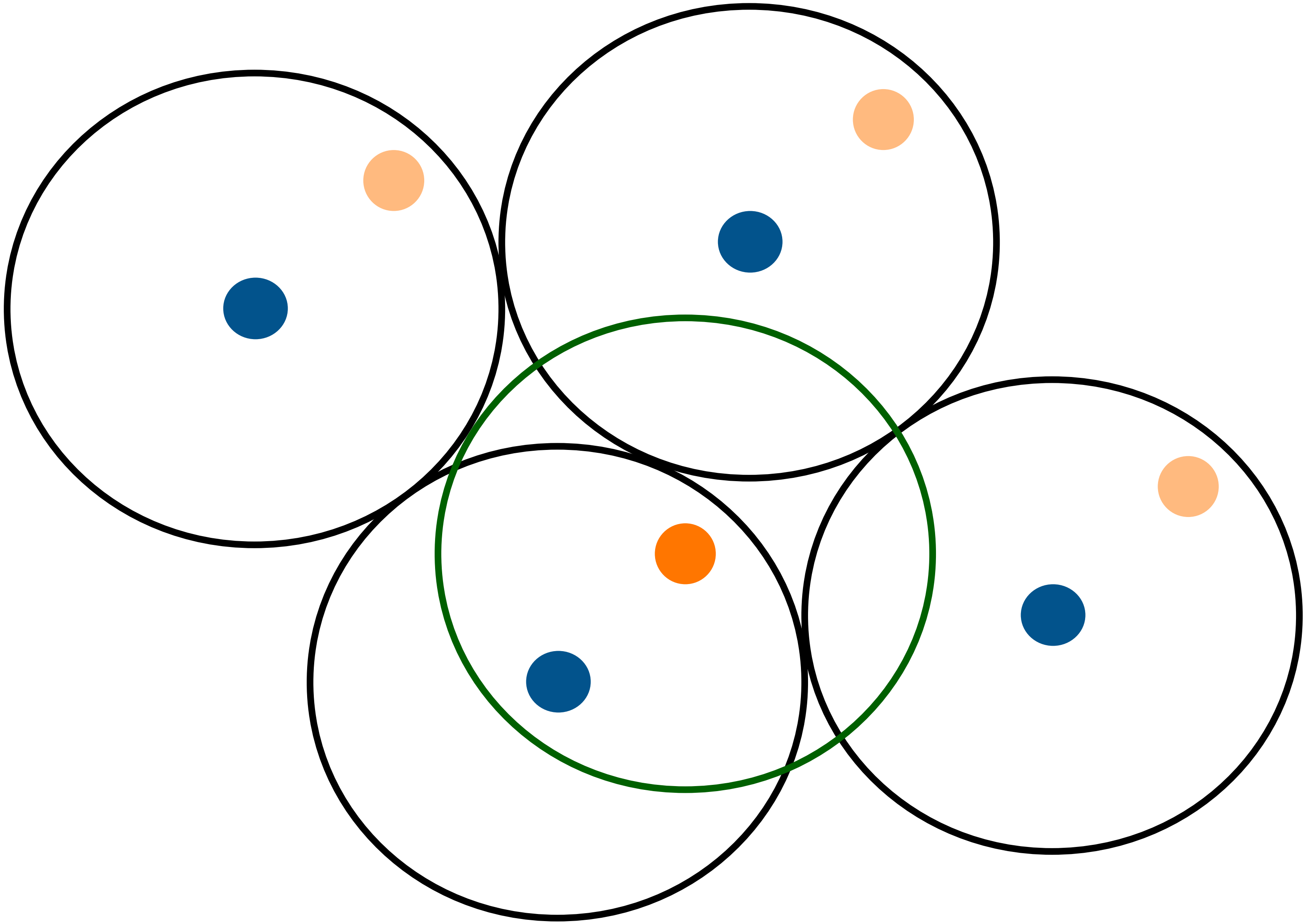
Error Correction

Example (Repetition Code)

$$0 \to 000$$
$$1 \to 111$$



Error Correction



Information Reconciliation and Privacy Amplification

Now, let the remaining sifted key at Alice side be S_A and at Bob side be S_B.

1. Alice sends $x = synd(S_A)$ to Bob.
2. Bob computes $S'_B = corr(x, S_B)$.

Note that if $d(S_A, S_B) < \frac{d-1}{2}$, the error correction code guarantee that

$$S_A = S'_B.$$

Outline

1. Quantum Key Distribution
2. Information Reconciliation
3. Privacy Amplification

Randomness Extractor

Definition (randomness extractor)

Let $X \in \{0,1\}^n$ be a random variable. An extractor is a function $Ext: \{0,1\}^n \to \{0,1\}^m$ such that $Ext(X) \approx U_m$.

Randomness Extractor

Example.

Suppose we have a biased coin:

$$\Pr(Y = 0) = p$$
$$\Pr(Y = 1) = 1 - p$$

We design an extractor $Ext(Y_1, Y_2) = Y_1 \oplus Y_2$. Then,

$$\Pr(Ext = 0) = p^2 + (1 - p)^2$$
$$\Pr(Ext = 1) = 2p(1 - p)$$

If $\Pr(Y = 0) = \frac{1}{4}$, then

$$\Pr(Ext = 0) = \frac{10}{16}$$
$$\Pr(Ext = 1) = \frac{6}{16}$$

Randomness Extractor

Example.

Suppose we have a biased coin:

$$\Pr(Y = 0) = p$$
$$\Pr(Y = 1) = 1 - p$$

We design an extractor that check bit string pairwisely such that

00	throw away
01	output 0
10	output 1
11	throw away

Then,

$$\Pr(Ext = 0) = \Pr(Y_i Y_{i+1} = 01) = p(1 - p)$$
$$\Pr(Ext = 1) = \Pr(Y_i Y_{i+1} = 10) = (1 - p)p$$

Privacy Amplification

In QKD setting, what we really care is how many randomness of $X|E$ can be extracted?

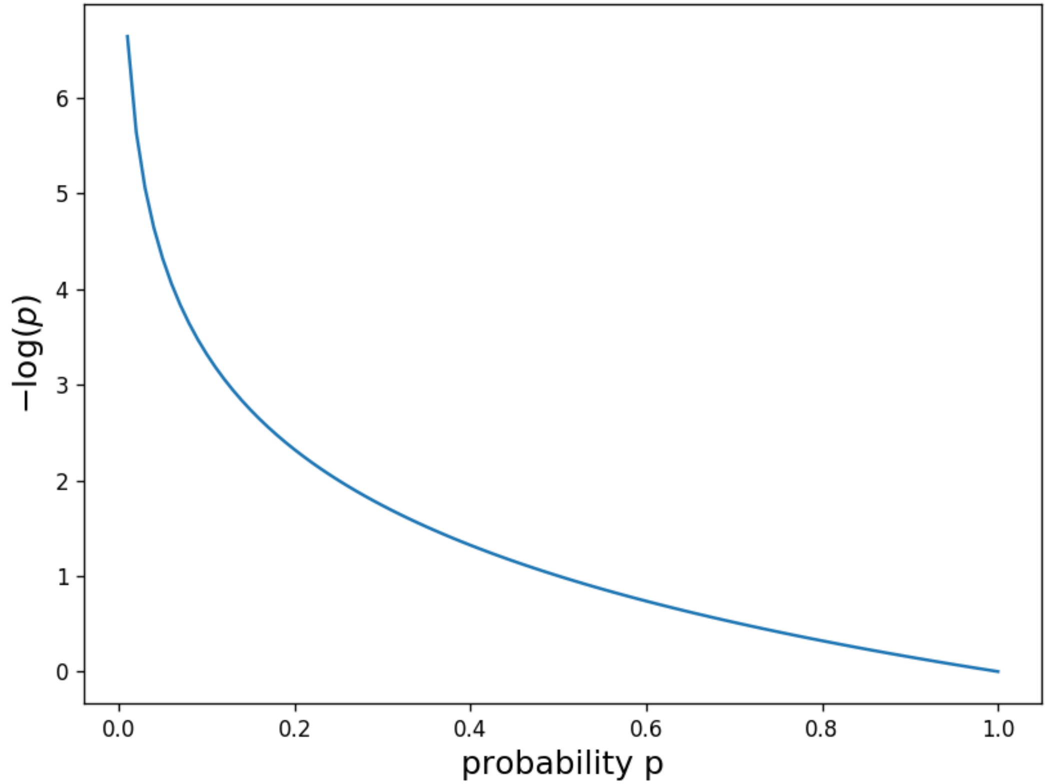
How can we quantify the randomness of a random variable Z?

First trial: Shannon entropy

$$H(Z) = -\sum_z p_Z(z) \log p_Z(z) \,.$$

Is Shannon entropy the right measure of the randomness in QKD?

Entropy



In entropic measure, we can view $-\log p$ as the amount of information.

Then, Shannon entropy $H(Z) = -\sum_z p_Z(z) \log p_Z(z)$ is the expectation of the amount of information we can get from Z.

Min-Entropy

Example (Shannon entropy)

Let Z be an n-bit string. Consider the following distribution:

$$p_Z(z) = \begin{cases} \dfrac{1}{2} & \text{, if } z = 11\cdots 1 \\ \dfrac{1}{2} \cdot \dfrac{1}{2^n - 1} & \text{, otherwise.} \end{cases}$$

When n is large, $H(Z) \approx \frac{n}{2}$.

The Shannon entropy is large. However, if we use Z as the secret key, Eve can find the plaintext with probability 1/2.

What‘s your best guessing probability?

♠ ♥ ♣ ♦

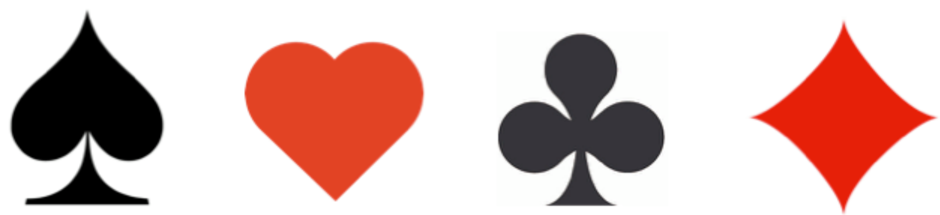
(1, 0, 0, 0)

What‘s your best guessing probability?

♠ ♥ ♣ ♦

(4/10, 3/10, 2/10, 1/10)

What‘s your best guessing probability?



(1/4, 1/4, 1/4, 1/4)

What‘s your best guessing probability?

For a gambler,
your only care about the choice with largest probability

♠

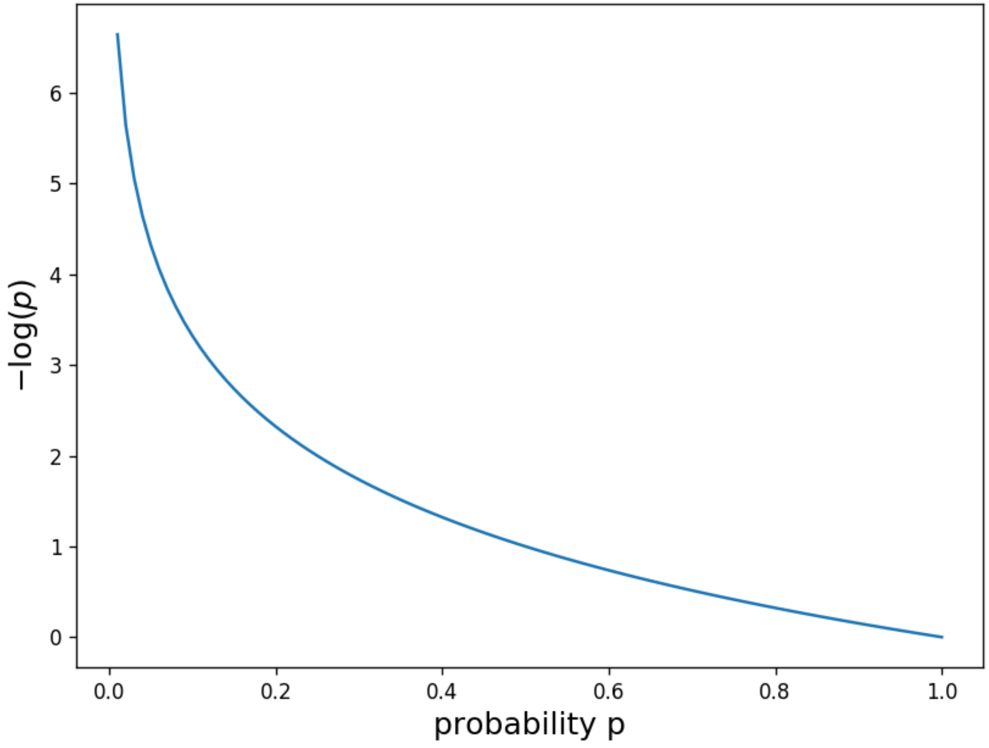
Min-Entropy

In cryptographic use, we mainly care about the mostly likely event (worst case) from the view of Eve.

Thus, the randomness of a random variable Z is the min-entropy

$$H_\infty(Z) \coloneqq \min_z - \log p_Z(z) \, .$$

Given $H_\infty(Z) = k$, leftover hash lemma guarantees that there exists a extractor such that output $k - 2\log\left(\frac{1}{\epsilon}\right)$ bits.



Privacy Amplification

1. Alice and Bob share a weak secret X, which may has some correlation with Eve.
2. Alice chooses a hash function h_i function from $\mathcal{H}$. She set $K_A = h_i(X)$ and announce h_i.
3. Bob computes $K_B = h_i(X)$.

In the end, K_A and K_B are the shared secret keys of Alice and Bob.

Conclusion

The security of QKD counts on

- When does the parameter estimation fail? What's the probability?
- How can we lower bound the min-entropy of the sifted key?