About the CODE

Why JAVA ?

Portability

UNIX, WINDOWS...

Simplicity

Graphical user interface, Server...

University

OOP, Team sync...

Our project :

Implementation :

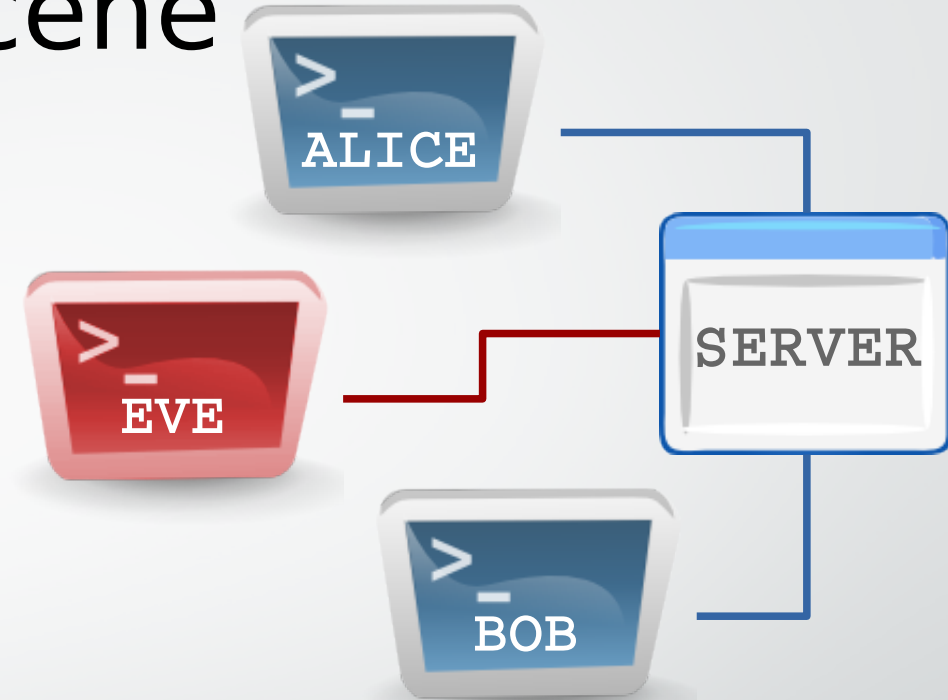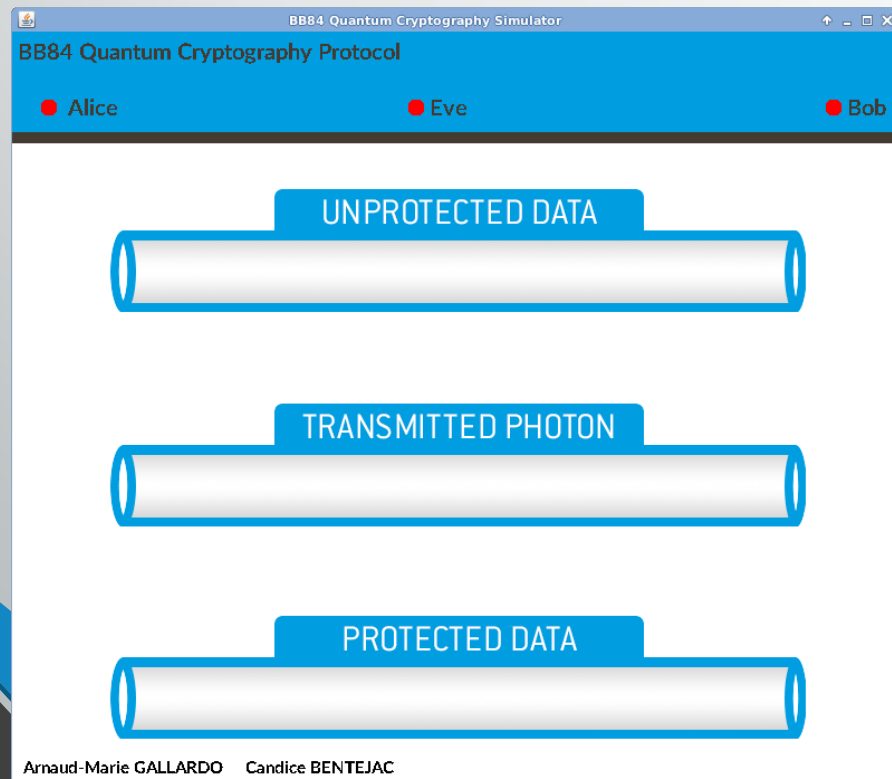**Server**

Graphical User Interface

Quantum Physics

Cryptography/BB84

▽ Total Lines of Code	**1618**
▽ src	**1618**
▽ (default package)	**1618**
Panel.java	307
Benchmark.java	271
BytesScheme.java	167
ServerCore.java	144
Window.java	120
Tests.java	120
Crypt.java	71
QuantumConnectionListener.java	67
FilterScheme.java	65
Filter.java	55
PhotonScheme.java	55
Main.java	40
Photon.java	37
BenchmarkGraph.java	34
Polarization.java	18
AbstractScheme.java	17
DebugConnectionListener.java	14
Basis.java	11
Scheme.java	5

Behind the scene

1 Client/Server
We used a library named JEXXUS, based on the Java Sockets API

ALICE

EVE

SERVER

BOB

BB84 Quantum Cryptography Simulator

BB84 Quantum Cryptography Protocol

● Alice   ● Eve   ● Bob

UNPROTECTED DATA

2 GUI (Graphical User Interface)
We used the Java Swing API

TRANSMITTED PHOTON

PROTECTED DATA

Arnaud-Marie GALLARDO   Candice BENTEJAC

Quantum physics implementation

FAKE TUNA
(pun intended)
Photons.java

POLARIZATION
Polarization.java

BASIS
Basis.java

FILTER
Filters.java

POLARIZED PHOTON
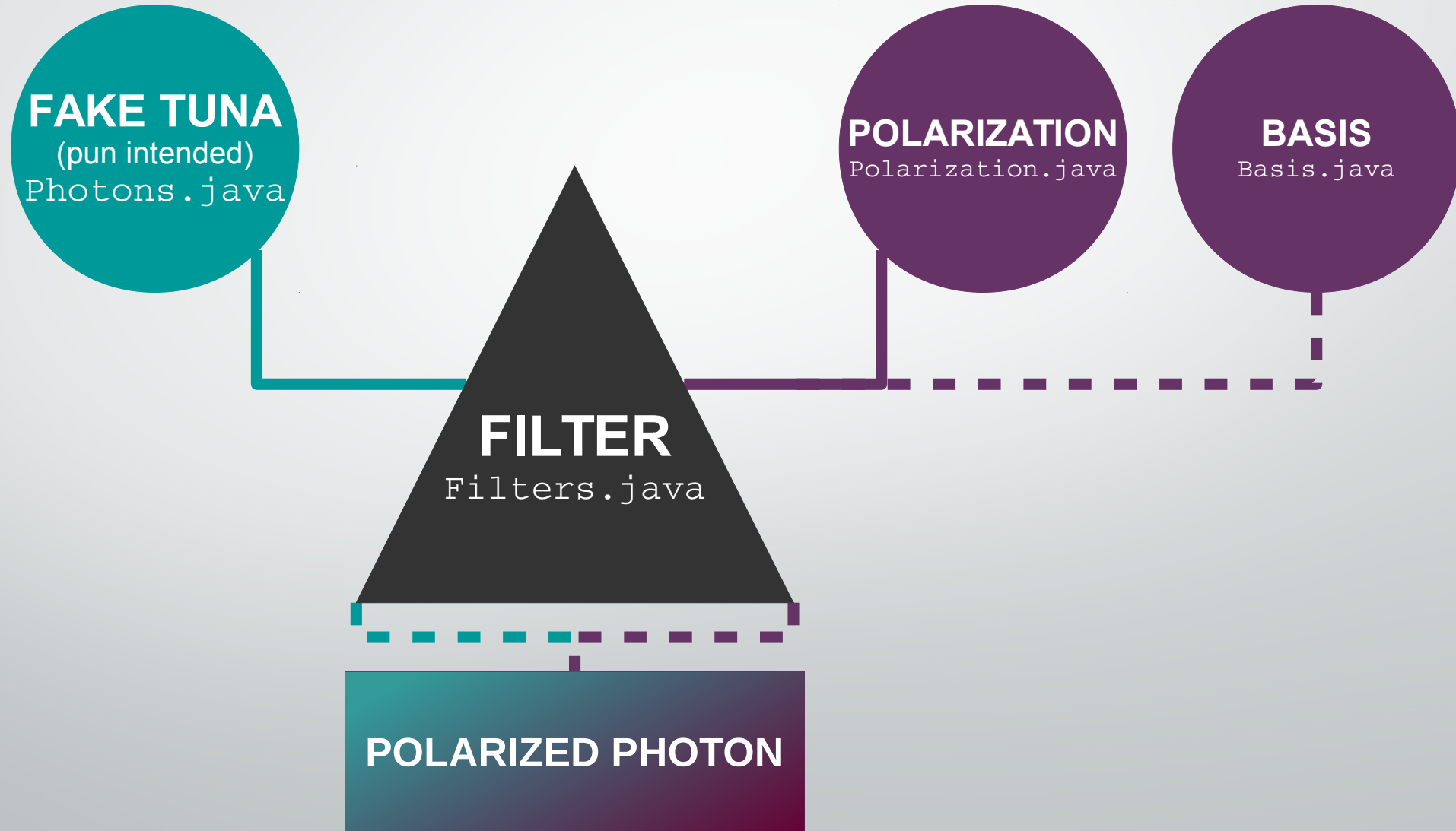
Team work
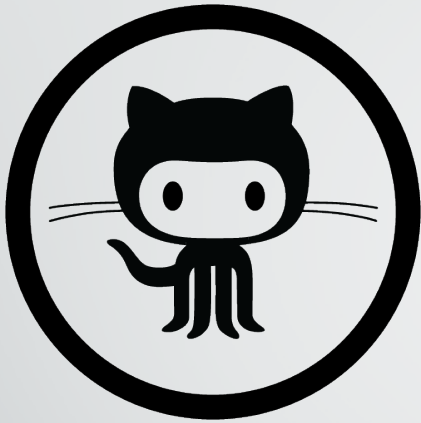
www.github.com/ArnaudGallardo/BB84

www.github.com/ArnaudGallardo/BB84-client

Benchmark

Apache POI - the Java API for Microsoft Documents

- Creating spreadsheets files

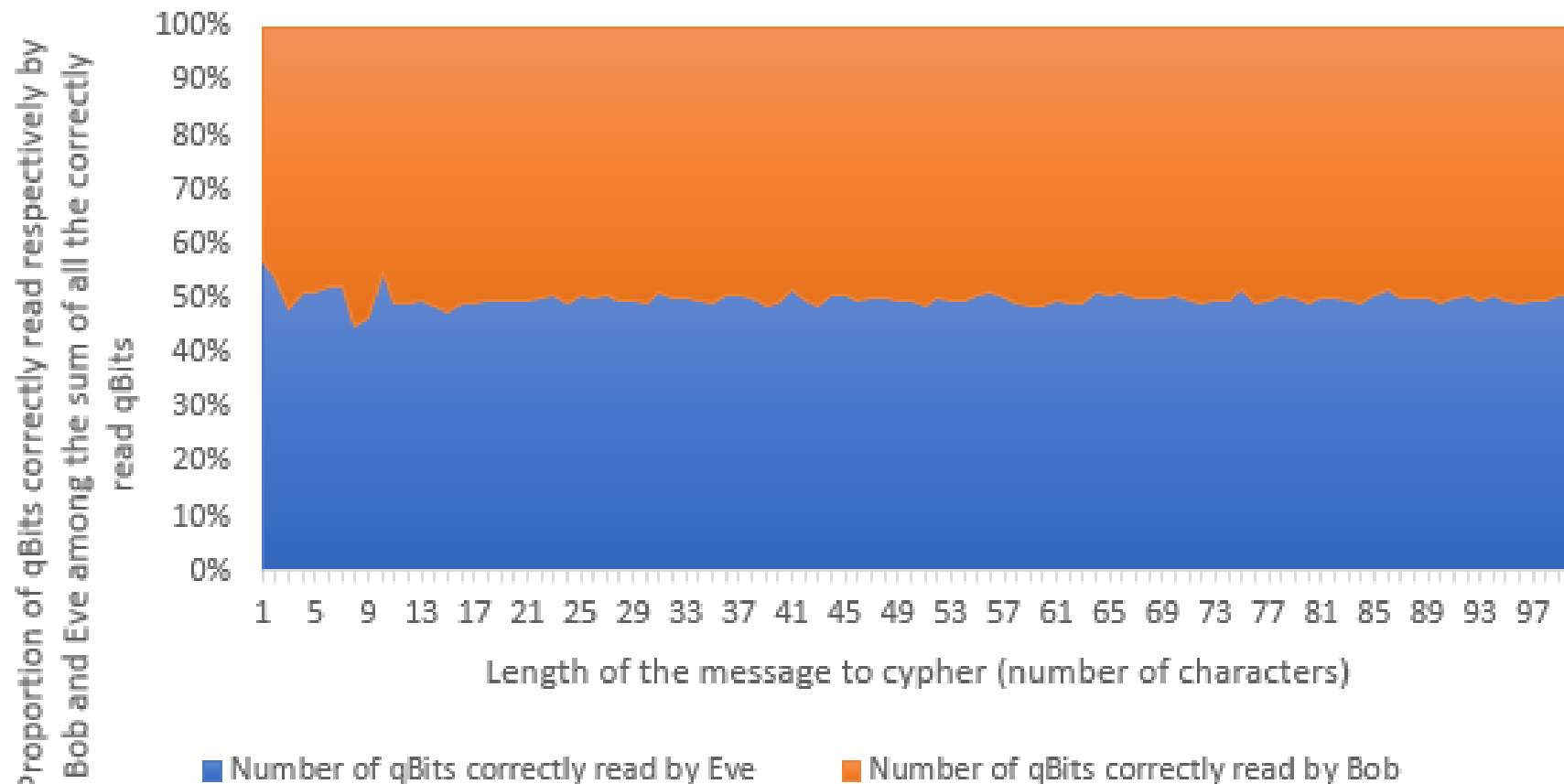- Automatic simulation (21000 tests in 80 minutes with at the end more than 16 millions photons simulated) and writing

Benchmark

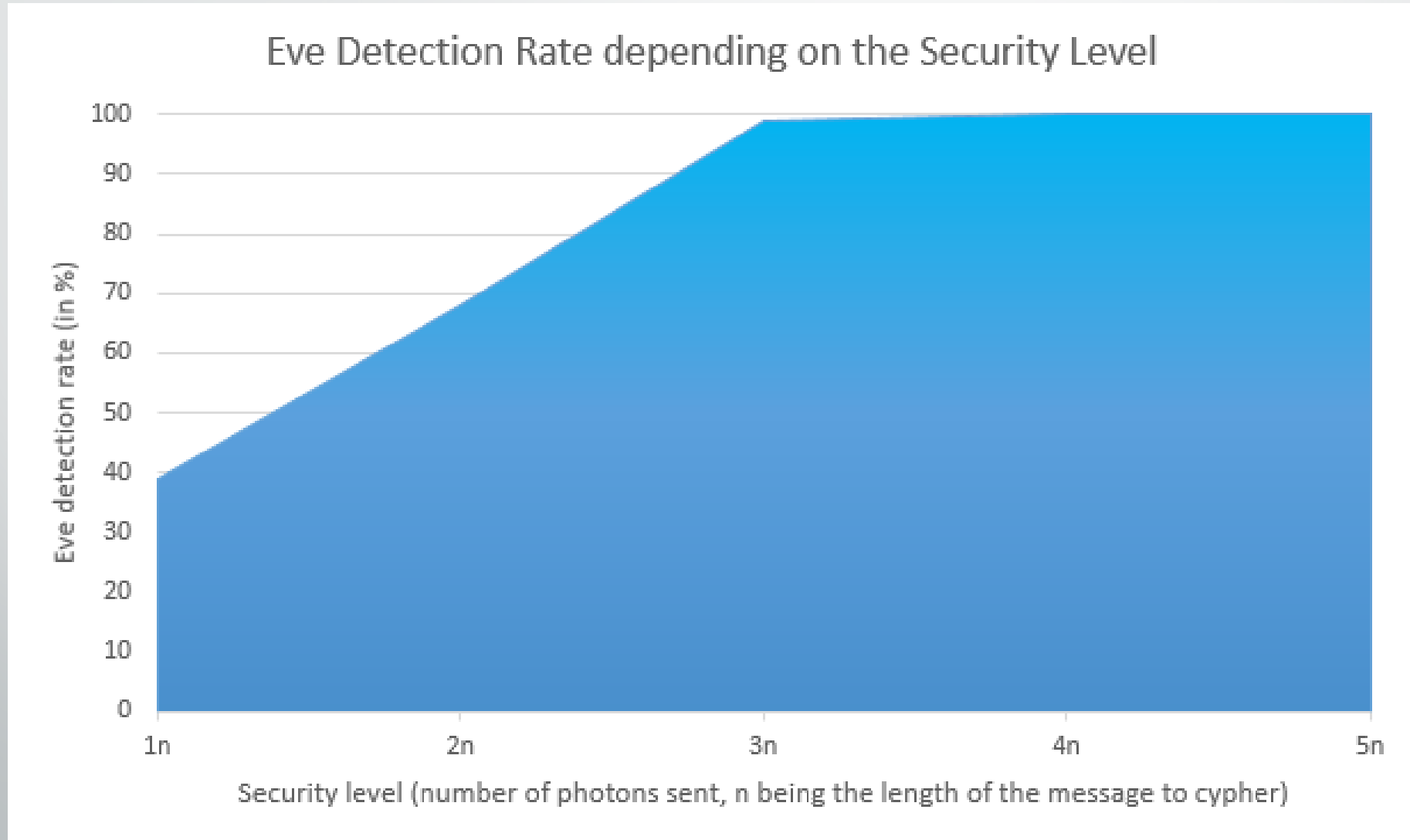| Number of Qbits sent by Alice	Number of Qbits correctly read by Eve	Number of Qbits correctly read by Bob	Number of sacrificed photons	Eve's detection
16	7	7	1	0
32	13	22	6	1
48	27	25	1	0
64	29	27	1	0
80	38	31	1	1
96	51	44	1	0
112	62	66	10	1
128	60	61	1	1
144	85	71	1	0
160	89	75	1	0
176	91	95	7	1
192	93	86	1	1
208	111	108	4	0
224	122	124	12	1
240	125	129	9	1
256	124	129	1	1
272	134	137	1	1
288	142	136	1	0
304	178	156	4	1
320	160	157	1	1
336	160	169	1	0
352	165	163	1	0
368	173	179	1	1
384	179	187	1	1
.	.	.	.	
.	.	.	.	.
.	.	.	.	.

Benchmark

Comparison between the qBits correctly read by Eve and the qBits correctly read by Bob

Benchmark

Eve Detection Rate depending on the Security Level

Benchmark

Eve Detection Rate with a 2^n Security Level

Eve detection rate (in %)

100
80
60
40
20
0

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Length of the message to cypher (number of characters)

■ Eve detection rate (average detection on 100 tests)