

## BB84: Write a hidden secret in the quantum world

Ohad Forman<sup>1,2</sup>, Noam Paryanti<sup>1,3</sup> and Georgi Gary Rozenman<sup>1,2</sup>

<sup>1</sup>Raymond and Beverly Sackler School of Physics and Astronomy,

Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel

<sup>2</sup>School of Electrical Engineering, Iby and Aladar Fleischman Faculty of Engineering, Tel Aviv University, Tel Aviv 69978, Israel

<sup>3</sup>School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel

(Dated: June 4, 2022)

We observe the results of the BB84 protocol Analog using a pulsed laser signals as the hardware systems that required for QKD. A simulation results of the chain of communication, the probabilities of Errors from Eavesdropper and the probabilities to detect Eavesdropper with respect to the Test string size is shown. We show how A GUI based on the simulation can deploy and help a "End user" understand and use the protocol for encrypted communication without deep knowledge in Quantum physics and mathematics and can use it when the technology will be available for the "end user"

### I. INTRODUCTION

Cryptography, from Ancient Greek can translate to "To write a hidden secret", since ancient times the need to secure information and choose the authorized sides to the information was essential in a way that a unauthorized side can't accesses the hidden information. The definition of Cryptography is the constructing and analyzing of protocols that allow to two or more authorized sides to exchange information without the ability of unauthorized side to get access to the information. In this era the transformation of secret data is essential to mankind. health, financial and state security data is only a part of the hidden information that need to be encrypted for a safe communication, in this era most of the communication is online and also stored in a cloud based technology. The Quantum World meets the Cryptography World in two main roots. One root is the Evesdropper root, where in the close future a Evesdropper will be able to evedrop a encrypted information with the help of Quantum computer[1]. In the side of the authorized sides of communication, The BB84 protocol introduced by Bennett and Brassard in 1983 [2] make a description of how to use rules of the Quantum mechanics to the purpose of generate a Quantum Key distribution protocol.

At 1900 quantum hypothesis made by Max Planck[3]. that any energy-radiating atomic system can theoretically be divided into a number of discrete "energy elements" such that each of these energy elements is proportional to the frequency with which each of them individually radiate energy.

5 Years later at 1905, with the motivation to explain the photoelectric affect, A.Einstein made the hypothesis that the light itself is made of individual quantum particles[4], which in 1926 came to be called photons by Gilbert N. Lewis[5]. The state in quantum physics is the mathematical description of the quantum physical properties of a quantum in the shape of the probabilities to measure the outcomes of each possible measurement on a system.

In 1939 P. A. M. Dirac introduce a new notation to quantum mechanics[6], Based on the Bra-Ket. One can say that a notation is only a notation, but Dirac notation

made the quantum state concept wider and dipper in the understanding of the quantum state nature.

Jumping to 1970, J.L.Park build the foundations to the "no cloning theorem" [7]. the no-cloning theorem states that it is impossible to create an independent and identical copy of an arbitrary unknown quantum state.

With this principals from the quantum mechanics world the BB84 protocol is able to solve a security problems of the classical encryption algorithms. The BB84 algorithm using the "no cloning theorem" in the shape of testing for the existing of a evesdropper, if a evesdropper is able to copy the encrypted information without changing it he cannot be discover, this problem has no solution in the classic world, this is the main problem that being solved with the BB84 protocol. another problem in the classical world is the difficulty in making a random Bit, that is essential to encryption, the states in quantum mechanics can be a superposition of 2 states with 0.5 probability to measure each one, by that this is a true random Bit generator if we call one of the states '0' and the second '1'.

The protocol will hold secure in the era of Quantum computers, since it is not depend on the computing power of the evesdropper and depend only on the principles of the quantum mechanics world.

### A. Random in technology

random result of a process is a result sample that distribute equally in the result field, assume that the results field is [0,1], as a binary bit. a random result is a function that return 0 or 1 with equal probability's 0.5. achieving such a function is not trivial, a computer software depends on a defined set of actions to deliver a result, an algorithm. A can be only "pseudo random" and breathable because of that, real randomness is non invariable function, and a "pseudo random" is reversible, one can try and succeed to understand the way the algorithm operate.

In the nature we witness a random phenomenons, in the Quantum world, particle decay process's and more, we



can use this phenomenons to achieve a ideally random generator.

#### B. One Time Pad

The One Time Pad is an Classical encryption method. the method is to generate a security key with logical binary bits, and share the key secretly between authorized sides of the communication, the key binary string must be random sequence, now a message with the same length is coded in binary representation. A Binary operator of 2 bits applies to a key bit and a massage bit result a random result also that the sender can sent safely, the receiver can apply the same logical operator with the result and the key to get the un-encrypted message XOR operator meet the condition for that operator with the Truth table:

BIT A	BIT B	XOR(A,B)
0	0	0
0	1	1
1	0	1
1	1	0

Table I. XOR truth table

fully secure in principle precede can achieve under 3 condition:

- The key is fully random
- The key can be used for one time only
- The key known only to authorized sides of communication

## C. Photon Polarization

The mane protocol in quantum cryptography - BB84 requires from the information bits to have some quantum properties. Since polarized Photons are stable, a practical way of transmission and they have the right quantum properties to execute BB84, they are the mane implementation of the protocol.

When classical polarized light is polarized along the X axis and then passes through a half-wave plate where the angle between the polarization of the electromagnetic wave and the mane axis of the half-wave plate is  $\theta$  the polarization is rotated by  $2\theta$ , as described in figure 1.

In photons there is a similar phenomenon, when a photon passes through a half-wave plate were the angle between the polarization of the photon and the mane axis of the half-wave plat is  $\theta$  the polarization is rotated by  $2\theta$ . So one can use a half wave plate to rotate the polarization of a linearly polarized photon. Beam splitter is a device that lets light to pass through it or to be reflected based on the polarization of the light. For example a beam splitter can ignore linearly polarized light in some

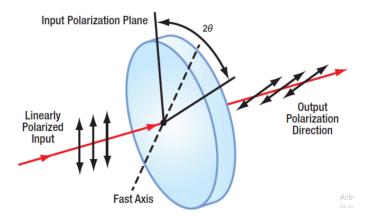


Figure 1. visual description of half-wave plate

angle we can define as 0, and completely reflect linearly polarized light in angles 90,-90. In this case a photon with some polarization angle  $\theta$  will pass with a probability of  $\cos(\theta)^2$  or be reflected otherwise, this is similar to Malus' low for polarizers.

#### D. Measurement Operators - Dirac notation

Armed with a basic understanding of photon polarization we can write a mathematical description of the phenomenon.

denote the state  $|\theta\rangle$  as the state that represents the polarization in the direction of  $\theta$ . Generally a state  $|\theta\rangle$  can be described as a super position of a base with two orthogonal angels, for example -45°,45° and 0°,90°. We know that they are in fact orthogonal because the polarization vector of a photon polarized in the direction of one angle does not have a component in the direction of the second one:

$$< -45|45> = < 0|90> = 0$$
 (1)

Because the probability of a polerized photon to pass though a beam splitter (or measuring it in one specific base) is controlled only by the angle of the polarization. The transition between the bases is given by:

$$|0\rangle = \frac{1}{\sqrt{(2)}}|45\rangle + \frac{1}{\sqrt{(2)}}|-45\rangle$$
 (2)

$$|90> = \frac{1}{\sqrt{(2)}}|45> -\frac{1}{\sqrt{(2)}}|-45>$$
 (3)

the 0,90 base is called "+" and the -45,45 base is called "X".

To measure the the different states we can define the following operators:

$$|M_x>=|45><45|-|-45><-45|$$
 (4)



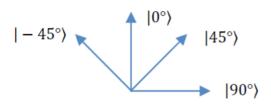


Figure 2. visual description of the 0-90 plane

$$|M_{+}\rangle = |0\rangle < 0| - |90\rangle < 90|$$
 (5)

So measuring a state in the same base as the operator has the same result every measurement, but measuring in a different base results a random result every time. This is the most important quantum property in BB84 because all of bob's (and eve's) measurements are done by choosing between  $M_x$  and  $M_+$  and 0 is represented by 90 or 45 and 1 is represented by 0 or -45 depending on the base that was chosen.

Bob will use a system of a beam splitter and a half-wave plate to implement  $M_x$  and  $M_+$  he would let the photon pass the half-wave plate to measure in the 0,90 base and rotate the polarization to measure -45,45.

#### E. BB84 Protocol

BB84 Protocol[2] is the first public distributing key protocol that use the Quantum mechanics principals to made a safe exchange of a secret key that can be used with the One Time Pad method. If the protocol execute in order and with the appropriate technology(Appendix: technologies) is fully secure and can detect a Evesdropper, what can't be done in classic protocols.

## 1. BB84 Protocol steps

### Eve steps, as a representation of a Evesdropper activity

- 1. Alice and Bob generate a random bits array in the size of the binary representation of the key+test+safe margin wanted for the bases. Alice generate a array in the same size for the bits Eve generate the same as Alice
- 2. Alice code the bits in the different bases (+ or x) as every photon represents one bit.

  Eve measure the state sent by Alice, with the bases generated. Eve code the bits in the bases and send this state with a single photon.

  Bob measure the state sent by Eve
- 3. Alice and Bob publicly compare the bases, discard bits with different bases and save bits with the same

bases.

## Eve also intercept this public information

- 4. Alice and Bob chose a size of test string and the location in the array of binary data to compare. if the comparison test result in different bit the protocol is stooped and Eve is discovered. If the there is no mismatches Alice proceed to the next step.
- 5. Alice send the message with One Time Pad method Eve intercept the encrypted message, now she have a information about the bases chosen that was matched between Alice and Bob and can use algorithms to brake the encryption and not discover by Alice and Bob.

Bob received a message that can be gibberish or Alice message, but in the two cases not encrypted (after decryption using One Time Pad method)

If Eve is not present the test results show that. Alice and Bob use the key generated in One Time Pad method safely.

## 2. BB84 Protocol Probabilities

We define events of the protocol as follow:

ESB := Eve chose a base equal to Alice BSB := Bob chose a base equal to Alice

If Eve chose the same base as Alice there will be no change in the state and Eve is not detectable

ESB as a probability of 0.5 BSB as a probability of 0.5 and independent

The Event that correspond to mismatch between the bits of Alice and Bob when chose the same base is the important event, in this event Eve can be discovered.

$$OBE := P(\overline{ESB}, BSB) = \frac{1}{2} \cdot \frac{1}{2} = 0.25 \tag{6}$$

From that the probability of mismatch in the bits where the bases are equal is 0.25 Now we define the event successful discovery of Eve in the test step:

ST := test is success in finding Eve when present

The probability to NOT find Eve in the test step is the independent events where of every test bit is the event  $\overline{OBE}$ . the probability to NOT detect Eve is the Event



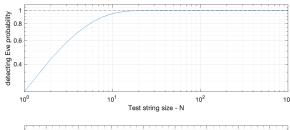
where every test bit is match between Alice and Bob

$$\overline{ST} := P(\overline{OBE})^N$$

where N = Test string size

The complimentary event is the event of Eve being detect, we defined this probability as a confidence level in the test step:

Confidence level := 
$$ST = 1 - P(\overline{OBE})^N = 1 - 0.75^N$$
(7)
let CF(N) be the confidence function as function of N



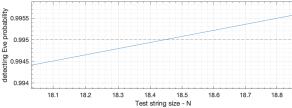


Figure 3. CF(N) - a Y=0.995 line (- -) represent a confidence level of 0.995, CF(N) squared up meet this level at N=19. CF converge relative "fast" to 1, where relative "fast" is considered with test string that make the computing "expensive" in time and resources

## II. EXPERIMENT CONFIGURATION

The hardware of the experiment is A pulsed laser transmitter, photons detectors, beamsplitter, Half wave plate. We introduce their symbols (Table II). and represent the experiment configuration in a block diagrams.

- Pulsed laser produces a linearly polarized photons pulses The initial quantum state.
- Half wave plate rotate the polarization the rotation of the plate is half of the polarization rotation (Appendix-Half wave plate)
- Beam splitter detect and distinguish a super position of the state, if the pulsed laser passes the half wave plate with a wrong bases the beamsplitter will

- transmit 0.5 (ideally) of the photons to every detector. with the same bases all the photons will transmit to the right detector.
- Laser pulses detectors a system that detect photons, include 2 detectors, if one is getting photons a led will indicate that detection, if both detect light in approximately same intensity only 1 will indicate by the led.

Device	Symbol
Half Wave plate	
Pulsed laser	
Beam spliter	
Photons detecter	

Table II. Symbols of the Hardware of the experiment system

Transmission block include a pulsed laser and a half wave plate(Fig 4).

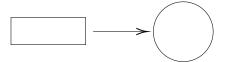


Figure 4. Transmitter Block

Detectors block include two photons detectors, half wave plate and beam splitter(Fig 5).

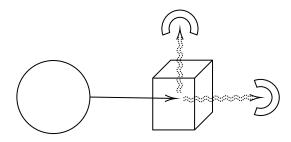


Figure 5. Detector block

### A. Communication without Eve

Communication without Eve is a transmittion of data from Alice and Bob detecting the data(Fig 6).



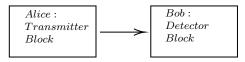


Figure 6. Alice and Bob Block diagram

#### B. Communication with Eve

Eve Detecting Alice data and transmit data to Bob, Eve is detecting and transmitting block. Eve placed between Alice and Bob(Fig 7).



Figure 7. Alice, Bob and Eve Block diagram

# C. differences between the experimental model and the theoretical model

Since sending a single photon at a time requires very expensive equipment, the experiment was done with classical light sent by a laser. the system is identical to the theoretical model except the change of the single photon source to a laser and because new when the wrong base is chosen the beam splitter splits the light to two different beams instead of choosing one path randomly like a single photon. Because we want to illustrate the case of a single photon when both detectors are activated a device will randomly choose a result.

## III. RESULTS

## A. Alice and Bob communication

the Experiment of Alice and Bob communication was done 3 times, once with 18 bits, once with 50 and once with 100. in every Experiment we calculated the number of times that Alice and Bob chose the same base. because the probability that Alice and Bob chose the same base is 0.5, the experiment's probability distribution can be described by a Binomial distribution, so the expected value for Alice and Bob to chose the same base in a N-bits message E(N) is:

$$E(N) = \frac{N}{2} \pm \frac{\sqrt{N}}{2} \tag{8}$$

the results of Alice and Bob communication for 18 bits are represented in table III and the results of all 3 messages are summed up in table IV  $\,$ 

Table III. the results of a message of 18 bits

alice base	alice msg	bob's base	bob's result
X	1	X	1
х	1	+	0
+	0	+	0
x	1	+	1
+	0	+	0
X	1	X	1
+	0	+	1
+	0	x	0
x	1	+	1
+	0	X	1
х	1	+	1
+	0	+	0
+	0	+	1
+	0	x	1
X	1	X	0
+	0	+	0
+	0	X	1
х	1	+	1

Table IV. alice and bob communication summary

Message length	Expected value	Disagriment percantage	Disagriment count	sigma	N sigma
18	9	50%	9	2.1	0
50	25	64%	32	3.5	2.0
100	50	67%	67	5.0	3.4
summary 18+50+100	84	64%	108	6.5	3.7

## B. Alice, Bob and Eve communication

the experiment of Alice, Bob and eve communication was done 3 times for the same messages length of the previous experiment. in every simulation we calculated the number of times that Alice and Bob chose the same base and eve was detected. because the probability that Alice and Bob chose the same base is 0.5 and the probability of detecting eve is 0.25, the detection of eve's probability distribution can be described by a Binomial distribution, so the expected value of detecting eve in an N-bits message  $E_{eve}(N)$  is:



$$E_{eve}(N) = \frac{N}{4} \pm \frac{\sqrt{3N}}{4} \tag{9}$$

the 18-bits full results are represented in table V where the bits with matched bases but eve was not detected are marked with yellow and the bits that detect eve are marked with red. a summary of all 3 measurements is represented in table IV with the simulation results.

Table V. the results of a message of 18 bits with eve

eve key	bob key	alice key	alice msg	eve msg	bob msg
+	x	+	0	0	1
Х	+	+	1	1	0
x	+	x	1	1	0
X	X	х	1	1	1
x	+	x	1	1	1
X	+	+	1	0	0
+	+	x	1	1	1
x	+	x	0	0	1
X	X	X	1	1	1
+	x	+	1	1	0
+	+	x	0	1	1
x	+	x	1	1	1
+	x	+	1	1	1
x	+	x	0	0	1
X	X	X	1	1	1
x	x	+	0	0	0
X	X	х	0	0	0
+	X	х	0	0	1

#### C. Simulation

Based on a BB84 protocol simulation, a MATLAB GUI as been made. the simulator GUI provide a information that not accessible to a "end user" that communicate via the GUI in real BB84 communication. to distinguish between the information that available we define for this section a "user" as a "end user" that will use the technology and will NOT be accessible to some of the information.

simulator Inputs is:

• Eve present [1 (for true), o (for false)] - real communication users CANT chose if Eve is present. this Input is to simulate the present of Eve, the Test results is the only indicator for the present of Eve for a user.

- Alice massage ["string"]
- Test string size [integer #] a figure that represent the probabilities to detect Eve as a function of the Test string size is part of the GUI design, such that a end user can choose without deep knowledge of the protocol what is a "safe" enough Test string size.

## The output is:

- Bob massage received ["string"] if the protocol didn't detect Eve the message received successfully if Eve is not present. if the protocol didn't detect Eve and Eve is present, from a bad choose of the Test string size, Bob will receive a "gibberish" message. if the protocol detect Eve present The message discard and a indicator string for that will be Bob received message.
- Number of Bits [integer #] The number of bits used are as follow Alice message length \* 7(Matlab default Binary array for a letter) \* 1000(for a large margin for test string size)
- Number of match bases of Alice and Bob [integer #]
- Number of match bits from the times that the bases are match [integer #] the number that indicate if Eve present, this number is NOT the test its purpose its academic and not part of the information that a User accessible to.
- Error [%] the percentage of mismatches in bits from the times that the bases match. also for academic use, the user is not accessible for this information.
- Test result if the test not finds Eve a indicator message with the confidence percentage % calculate from the probability to detect Eve with the test string size that the user chosen if the test finds Eve a string indicate Eve present if the test not finds Eve but Eve present, the test will indicate that Eve is not present. This is the information that available to the user, if the test string size is not large enough there is a possibility that Eve will NOT discover.

simulation result figures (8,9,10) show a simple GUI app that deliver the necessary information to a user that need only the knowledge about the test string size, figure 3 are built in the app, for help in this choice. the statistics of the confidence level are a Tool for the user, the rest are for academic purpose.



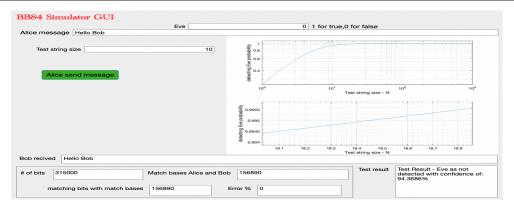


Figure 8. A communication between Alice and Bob when Eve is not present- Alice sends "Hello Bob", the size of the Test string as chosen to be 10. the number of bits/bases is 31500. The number of matched bases between Alice and Bob is 157381 and also the matching bits from the matching bases, from that the error is 0 that correspond to the chose of Eve not present. the test result output indicate that Eve is not present with a Laval of confidence of 94.3686%. Bob received the correct massage with high confidence "Hello Bob".

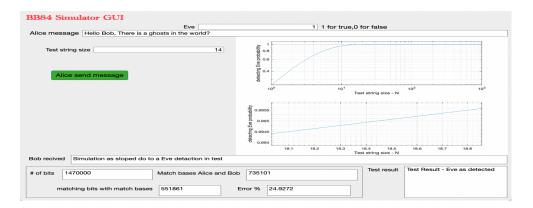


Figure 9. A communication between Alice and Bob when Eve is present- Alice sends "Hello Bob, There is a ghosts in the world?", the size of the Test string as chosen to be 14. the number of bits/bases is 1470000. The number of matched bases between Alice and Bob is 735101 and matching bits from the matching bases is 551851, from that the error is 24.9272 that correspond to the chose of Eve IS present. the test result output indicate that Eve is present. Bob received the indicator massage that a Eve is present, now Bob can say to Alice that something is standing between them in the line of communication, and Alice now know that there is a ghosts in the world

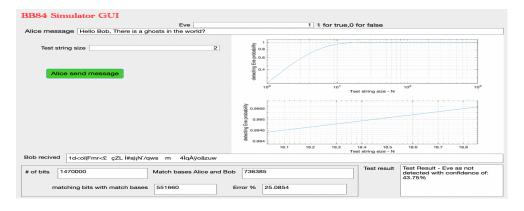


Figure 10. A communication between Alice and Bob when Eve is present- Alice sends "Hello Bob, There is a ghosts in the world?", the size of the Test string as chosen to be 2. the number of bits/bases is 1470000. The number of matched bases between Alice and Bob is 736385 and matching bits from the matching bases is 551660, from that the error is 25.0854 that correspond to the chose of Eve IS present. the test result output indicate that Eve is NOT present with confidence of 43.75%. Bob received a "gibberish" message.



Table VI. Lab results with Eve-Summery

Message length	expected matches	Expected detections value	detection percantage	num of detections	num of matches	sigma N sigma
18	9	4.5	11%	2	6	3.2   0.8
50	25	12.5	14%	7	30	5.3   1.0
100	50	25	18%	18	43	7.5   0.9
summary 18+50+100	84	42	16%	27	79	9.7   1.5

Table VII. Simulation results with/without Eve-Summery

Message length	Test size	Match Bases	Match Bases detected	Error %	Confidence level in test%	Eve present	Eve detected	Bob received correct message
315000	10	156980	156980	0	94.4	X	<b>X</b>	✓
1470000	14	735101	551869	24.9	$\rightarrow 100$	✓	<b> </b>	✓
1470000	2	736385	551660	25.1	43.75	✓	<b>X</b>	×

#### IV. CONCLUSIONS

the protocol was able to detect eve for chose of string size that statistic "big enough", in the case of the bad communication, case of row 3 in table (VII) the test string size chose to 2 with confidence level of 43.75 presence where correspond to a bad communication more than half of the encryption procedures, this show the importance of choosing the test size correctly, one can say that choose it as big as you want, but it will cost in computing time in big message. The detection percentage of eve was a little smaller then expected in the lab, this is because the random device had a tendency to choose one result over the other one (not random), fact that we observe in the laboratory, this may cause by higher sensitively of one detector over the other, or by not ideal angle of transmission that caused by human error. because the ratio of times eve could be detected

and was not detected is much less then 50 percent (when Alice and bob choose the same base and a different one from eve) the simulation went as expected without any unexpected results and had about 25 percent of detection of eve and about 50 present of the bases were the same between Alice and bob.

The simple interference GUI show that in the future the BB84 can help every user with no need in understanding the Quantum and mathematics principles behind it.

For now it already shown that BB84 protocol technology is exist, One photons detectors and transmitters are exist, and a experiment made by the Chinese using satellites as been made with successes.

The hardware is Large and expensive, minimize the hardware in the future will give the possibility to communicate safely for every user with a computer and phone, what can be necessary in the era of Quantum computers.

<sup>[1]</sup> K.Vishi, M.D.Zych, and A.Jøsang, The impact of quantum computing on present cryptography, arXiv preprint arXiv:1804.00200 (2018).

<sup>[2]</sup> C. G. S.Breidbart and S.Wiesner, Quantum cryptography, or unforgeable subway tokens., Advances in Cryptology, 267 (1983).

<sup>[3]</sup> M. Planck, On the law of distribution of energy in the normal spectrum, Annalen der physik 4, 553 (1901).

<sup>[4]</sup> A. Einstein, On the electrodynamics of moving bodies, Annalen der physik 17(10), 891 (1905).

<sup>[5]</sup> G. N. Lewis, The conservation of photons, Nature 118(2981), 874 (1926).

<sup>[6]</sup> P. A. M. Dirac, A new notation for quantum mechanics, Mathematical Proceedings of the Cambridge Philosophical Society 35(3), 416 (1939).

<sup>[7]</sup> J.L.Park, The concept of transition in quantum mechanics, Foundations of Physics 1 (1), 23–33 (1970).



	V. APPENDIX	18	Panel
			matlab.ui.container.Panel
	A. FULL MESURMANTS DATA	19	$\operatorname{ErrorTextArea}$
	A. FOLL MESORMANTS DATA		matlab.ui.control.TextArea
		20	ErrorTextAreaLabel
	https://drive.google.com/drive/folders/	20	
	1m5r0bjuwAIeUmEaCxibBIPQYE3ppbCax?usp=		matlab. ui.control.Label
	sharing	21	matching bits with match bases Text Area
			$\operatorname{matlab}$ . $\operatorname{ui}$ . $\operatorname{control}$ . $\operatorname{TextArea}$
		22	matchingbits with matchbases TextArea Label
			matlab.ui.control.Label
		23	ofbitsTextArea
	B. TECHNOLOGIES	23	matlab . ui . control . TextArea
	systems for quantum encryption:	24	ofbitsTextAreaLabel
			matlab.ui.control.Label
	http://www.idquantique.com/quantum-safe-crypto/	25	${\bf Matchbases Alice and Bob Text Area}$
	Random number generators :		$\operatorname{matlab}$ . $\operatorname{ui}$ . $\operatorname{control}$ . $\operatorname{TextArea}$
	http://www.idquantique.com/	26	${\bf Matchbases Alice and Bob Text Area Label}$
	random-number-generation/		matlab.ui.control.Label
	Single Photons transmitter and detectors	27	AlicesendmessageButton
	https://www.ape-berlin.de/en/	21	e e e e e e e e e e e e e e e e e e e
	quantum-dot-single-photon-generation-source/		matlab.ui.control.StateButton
	quantum-dot-single-photon-generation-source/	28	Image
			matlab.ui.control.Image
		29	end
	C. SIMULATION/GUI MATLAB CODE	30	
		31	% Callbacks that handle component
		51	events
1	classdef Sim_final < matlab.apps.AppBase		
2		32	methods (Access = private)
3	% Properties that correspond to app	33	
3	components	34	% Value changed function:
	•		${\bf Alicemess age Edit Field}$
4	properties (Access = public)	35	function
5	UIFigure		A licemes sage E dit Field Value Changed
	matlab.ui.Figure		(app, event)
6	BB84SimulatorGUILabel		value = app.
	matlab.ui.control.Label	36	
7	for true 0 for false Label		${ m Alicemessage Edit Field}$ .
	matlab.ui.control.Label		Value;
	TestresultTextArea	37	message = value;
8	matlab.ui.control.TextArea	38	$message\_length = strlength$
			message);
9	${\bf Testresult Text Area Label}$	39	end
	matlab.ui.control.Label	40	
10	${\bf Bobrecived Text Area}$	41	% Callback function
	$\operatorname{matlab}$ . $\operatorname{ui}$ . $\operatorname{control}$ . $\operatorname{TextArea}$		
11	${\bf Bobrecived Text Area Label}$	42	function EveSwitchValueChanged(
	matlab.ui.control.Label		app, event)
	EveEditField	43	value = app.EveSwitch.Value;
12		44	Eve_is_here = value;
	matlab.ui.control.	45	$\operatorname{end}$
	Numeric Edit Field	46	
13	${\tt EveEditFieldLabel}$	47	% Value changed function:
	matlab.ui.control.Label	41	TeststringsizeEditField
14	${\bf Alicemes sage Edit Field}$		
	matlab.ui.control.EditField	48	function
1.5	AlicemessageEditFieldLabel		Test string size Edit Field Value Changed
15			(app, event)
	matlab.ui.control.Label	49	value = app.
16	TeststringsizeEditField		TeststringsizeEditField.
	matlab.ui.control.		Value;
	${ m NumericEditField}$	E0.	Test_size = value;
17	TeststringsizeEditFieldLabel	50	· · · · · · · · · · · · · · · · · · ·
	matlab. ui. control. Label	51	end



Waste   Caliback function   BobrecivedTextAreaValueChangeds   (app, event)   so   value = app.   BobrecivedTextArea. Value; so   end   GetthemessageFuntantion   so   display   can   ca	52		86	%transmition bitween 1
BobrecivedTextAreaValueChanged	53	% Callback function		and 2 (Alice and Bob
(app, event)   value   app.   Eve.base(i)   Eve.Bits(i)	54		1	
Second   S			Cl 87	
BobrecivedTextArea.Value; so			88	. ,
## Alice_Bits(i) ## Callback function ## GetthemessagefromAliceButtonValueChanged (app, event) ## answer = 'what your want to 20 ## answer = 'what your want to 20 ## answer; ## app. BobrecivedTextArea. Value 20 ## app. event) ## app. BobrecivedTextArea. Value 20 ## app. app. BobrecivedTextArea. Value 20 ## app. app. app. app. app. app. app. ap	55			
Second   S	***		89	
### Scallback function		end		Affice_Bits(1)
Section   GetthemessagefromAliceButtonValueChanged (app, event)   Section   Gap, event   Gap, event   Section   Gap, event   Gap, event   Section   Gap, event   Section   Gap, event   Section   Gap, event   G		% Callback function	00	, also
GetthemessagefromAliceButtonValueChanged (app, event)  a naswer = 'what your want to as display'; a app. BobrecivedTextArea. Value = answer;  and display'; and end and 2 (Alice and Bob OR Alice and Eve) bor i=N.string if Bob.base(i) = Eve.base(i) and Eve.base(i) = Bob.Bits(i) = and ([0,1],1,1); and and 2 (Alice and Bob OR Alice and Eve) bor i=N.string and ([0,1],1,1); and cond and Eve.and if in the string of alice bot and Eve.base (i) = and Eve.and bob or and Eve and bits string and Eve and Bob bits(i) = and Eve.  Bob.Bits(i) = and Eve.				
(app, event)  answer = 'what your want to as display'; app. BobrecivedTextArea. Value aps. BobrecivedTextArea. Value as answer;  end  answer; app. BobrecivedTextArea. Value as and 2 (Alice and Bob of R Alice and Eve of For i=N.string if Bob.base(i) = Eve.Bits(i); app. down. AlicesendmessageButton valueChanged (app. event) and 2 (Alice and Eve) for i=N.string if Bob.base(i) = Eve.Bits(i); app. down. AlicemessageEditField. Value; as message.length = strlength( amessage); and and 2 (Alice and Bob of R Alice. Al	59			* /
answer = 'what your want to adisplay'; app. BobrecivedTextArea. Value so end  answer; answer; answer; answer; and and 2 (Alice and Bob Bits = zeros (N,1); wransmition bitween 1 and 2 (Alice and Bob OR Alice and Bob Bob.Bits (i) = Eve.base(i) = Bob.Bits(i) = Eve.base(i) in Bob.Bits(i) = Eve.base(i) in Bob.Bits(i) = Eve.base(i) in Bob.Bits(i) = Eve.base(i) in Bob.Bits(i) = Eve.Bits(i); and			araconangea	
### answer = 'what your want to ### app. BobrecivedTextArea. Value ### Bob.Bits = zeros(N,1);	60	(app, event)	92	_ · · · · · · · · · · · · · · · · · · ·
Second   S		answer = 'what your want to		
### app. BobrecivedTextArea. Value	01			
and 2 (Alice and Bob OR Alice and Eve)	62			
OR Alice and Eve)  *** Value changed function:	02		93	
Solution		· · · · · · · · · · · · · · · · · · ·		
65       % Value changed function:       67       if Bob_base(i) = Eve_base(i)         66       AlicesendmessageButton       88       Bob_Bits(i) = Eve_base(i)         66       AlicesendmessageButtonValueChanged (app., event)       99       else         67       Message = app. event)       99       else         68       message = app. alice = strlength( solution		end		,
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		Walue shanged function.		9
## Alice_Button	65		97	* *
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		<u> </u>		` '
67       message = app.       100       Bob_Bits(i) = randi ([0,1],1,1);         68       message.length = strlength( 101 message);       end       ([0,1],1,1);         69       Test.size = app. 103 message.length = strlength( 104 message);       end       end         69       Test.size = app. 103 message.length = strlength( 104 message);       else       else         70       Eve.is.here = app. 106 message.length = strlength( 104 message.length	66			` ′ , .
## Bob_Bits(i) = ## Alice_Bits = randi				
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		, ,		
Value; message length = strlength( 101 message); message); 102 end message); 103 Test.size = app. 105 Test.size = app. 106 Eve.is.here = app. 107 Eve.is.here = app. 108 for false , 1 for true 107 for i=N.string 12 %Number of initial random 108	67		100	` '
68       message_length = strlength ( 101 message); 102 message); 102 message); 102 message); 103 message.       102 message mend message.       end message mend message.         69       Test.size = app. 103 message.       105 message.       Bob_Bits = zeros(N,1); 105 message.         70       Eve_is_here = app. 106 message.       106 message.       %transmition bitween 1 message.         71       107 message.       for i=N.string message.       if Alice_base(i) = Bob_base(i) = Bob_base(i)         71       N= message.length*7*5e3 message.length*7*5e3 message.       109 message.length*7*5e3 message.       Bob_Bits(i) = Bob_Bits(i) = Bob_Bits(i) = message.         74       N_string = 1:1:N; message.       Alice_bits = randi (interval) message.       110 message.       Bob_Bits(i) = Bob_Bits(i) = message.         75       App. ofbitsTextArea.       Value= string(N); message.       110 message.<				
## Test.size = app.				
TeststringsizeEditField. 104 Value; 105 Bob_Bits = zeros(N,1);  70 Eve_is_here = app. 106 Eve_EditField.Value;% 0	68		101	
TeststringsizeEditField. 104			102	end
Value;  Eve.is_here = app. 106  Eve.EditField.Value;% 0 for false, 1 for true  Number of initial random 108 bits  N = message_length*7*5e3 109  Nostring = 1:1:N; app.ofbitsTextArea.Value= string(N); and Eve and bits string for Alice and Eve and bits string for Alice  Alice_base = randi([0,1],N) 114  Alice_base = randi([0,1],N) 115  Alice_base = randi([0,1],N,1); Bob_base;  Alice_base = randi([0,1],N,1); Bob_base;  If Alice_base(i) = Bob_base(i)  Alice_base = randi([0,1],N 114 end  Alice_base = randi([0,1],N,1); Bob_base = randi([0,1	69		103	
Eve_is_here = app.  EveEditField.Value;% 0  for false, 1 for true  72  %Number of initial random    bits  73  N = message_length*7*5e3  74  N_string = 1:1:N;  75  app. offitsTextArea.Value=    string(N);  76  %create base for Alice Bob    in    for Alice    and Eve and bits string    for Alice  79  Alice_base = randi([0,1],N)  79  Alice_base = randi([0,1],N)  80  Alice_Bits = randi([0,1],N,1);  81  Bob_base = randi([0,1],N,1);  82  Bob_base = randi([0,1],N,1);  83  Bob_base = randi([0,1],N,1);  84  if Eve_is_here = 1  119  %bits of match base  Eve_Bits = zeros(N,1);  100  %transmition bitween 1  and 2 (Alice and Bob  OR Alice and Eve  OR Alice_base(i) =  Bob_base(i)  Alice_Bits(i) =  randi  ([0,1],1,1);  end  end  end  Alice and Bob  si blue and Bob  si b			104	
EveEditField.Value;% 0 for false, 1 for true  71			105	
for false , 1 for true  for is Number of initial random bits  N = message_length *7*5e3  N = message_length *7*5e3  app.ofbitsTextArea.Value= string(N);  and Eve and bits string for Alice  Alice_base for Alice  Alice_base = randi([0,1],N)  Alice_Bits = randi([0,1],N,1);  Bob_base;  Alice_base = randi([0,1],N,1);  Bob_base;  Alice_base = randi([0,1],N,1);  Bob_base = randi([0,1],N,1);  Bob_base = randi([0,1],N,1);  Bob_base = randi([0,1],N,1);  The second of th	70		106	
71				· · · · · · · · · · · · · · · · · · ·
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		for false, 1 for true		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	71		107	~
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	72		108	
<pre>N_string = 1:1:N; app.ofbitsTextArea.Value=</pre>				· · ·
75       app.ofbitsTextArea.Value=       ;       string (N);       110       else         76       %create base for Alice Bob       111       Bob_Bits(i) =         80       and Eve and bits string for Alice       ([0,1],1,1);         80       Alice_base = randi([0,1],N       114       end         91       Alice_Bits = randi([0,1],N       115       %find matching bases bitween         10;       Alice and Bob       halice and Bob       halice and Bob       halice and Bob         81       Bob_base = randi([0,1],N,1);       117       index_equal_base=Alice_base       halice_base = Alice_base         82       Eve_base = randi([0,1],N,1);       ==Bob_base;         83       118       %bits of match base         84       if Eve_is_here == 1       119       %bits of match base         85       Eve_Bits = zeros(N,1);       120       Alice_Bits_same_base =	73		109	$Bob_Bits(i) =$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	74	- · · · · · · · · · · · · · · · · · · ·		Alice_Bits(i)
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	75	app.ofbitsTextArea.Value=		;
$\begin{array}{cccccccccccccccccccccccccccccccccccc$			110	else
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	76	%create base for Alice Bob	111	$Bob_Bits(i) =$
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		and Eve and bits string		randi
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		for Alice		([0,1],1,1);
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	77	% pseudo - random	112	$\operatorname{end}$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	78		113	$\operatorname{end}$
Alice_Bits = randi([0,1],N 116	79	$Alice\_base = randi([0,1], N$	114	$\operatorname{end}$
$,1); \\ Bob\_base = randi([0,1],N,1); \\ 117 \\ Eve\_base = randi([0,1],N,1); \\ 83 \\ 84 \\ if Eve\_is\_here == 1 \\ Eve\_Bits = zeros(N,1); \\ 119 \\ 119 \\ 120 \\ Alice\_Bits\_same\_base = 1 \\ 119 \\ 120 \\ Alice\_Bits\_same\_base = 1 \\ 119 \\ 120 \\ 120 \\ 120 \\ 120 \\ 130 \\ 140 \\ 150 \\ 140 \\ 150 $		,1);	115	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	80	$Alice_Bits = randi([0,1], N$	116	%find matching bases bitween
$\begin{array}{cccccccccccccccccccccccccccccccccccc$				
Eve_base = randi( $[0,1],N,1$ ); =Bob_base;  if Eve_is_here = 1 119 %bits of match base  Eve_Bits = zeros(N,1); 120 Alice_Bits_same_base =	81		117	index_equal_base=Alice_base
if Eve_is_here == 1	82			
if $Eve\_is\_here == 1$ 119 %bits of match base $Eve\_Bits = zeros(N,1)$ ; 120 Alice_Bits_same_base =		(1 / 1 / / /	118	,
Eve_Bits = $zeros(N,1)$ ; Alice_Bits_same_base =		if Eve_is_here == 1		%bits of match base
				Alice_Bits_same_base =



	<pre>index_equal_base);</pre>		Test_size) *100)
21	• , ,	148	disp ('message is being
.22	$Bob_Bits_same_base =$		encryipt with the key
	Bob_Bits(index_equal_base		')
	);	149	binary = reshape(dec2bin
23			(message, 8).'-'0'
24	$index_eq_bits =$		$,1\;,[\;]\;)\;;$
	Alice_Bits_same_base==	150	
	Bob_Bits_same_base;	151	$if Test\_size == 0$
.25		152	${\tt encriped\_Alice\_messege}$
.26	%precence of match		= xor(transpose(
.27	$sum_eq_bases = sum($		Alice_Bits_same_base
	index_equal_base)		(1: length (binary)
.28	$sum_eq_bits = sum($		)), binary);
	index_eq_bits)	153	
.29	$Pre\_of\_match\_bits = ($	154	else
	sum_eq_bases	155	transpose (
.30	$-\operatorname{sum}_{-\operatorname{eq}_{-\operatorname{bits}}})/$		$A$ lice_Bits_same_base (
	$sum_eq_bases*100$		Test_size: Test_size +
.31			length(binary)-1));
.32	app.	156	encriped_Alice_messege =
	matchingbitswithmatchbase	sTextArea	xor (transpose (
	. Value = string (		Alice_Bits_same_base (
	$\operatorname{sum}_{-\operatorname{eq}_{-\operatorname{bits}}});$		Test_size: Test_size +
.33	app.	<b>A</b>	$\frac{\text{length}(\text{binary})-1)}{1}$ ,
	Matchbases Alice and Bob Text	Area	binary);
	. Value = string (	157	disp('encrypted message
	$\operatorname{sum}_{-\operatorname{eq}_{-\operatorname{bases}}}$ );		as sent to Bob')
.34	app.ErrorTextArea.Value =	158	$\operatorname{end}$
	string(Pre_of_match_bits)	159	else
	;	160	disp ('Test Result – Eve
.35	OV TO		as detected')
.36	%Test	161	app. TestresultTextArea.
.37	$Test\_Alice =$		Value = 'Test Result
	Alice_Bits_same_base (1:		- Eve as detected';
	$Test\_size$ ); $Test\_Bob =$	162	disp('The protocol as
.38			stoped')
	Bob_Bits_same_base(1:	163	end
	Test_size);	164	Who dearing Alice maggage
.39	test_result_array =	165	%Bob decript Alice message
	Test_Alice=Test_Bob;	166	$ \begin{array}{ccc} \text{if } & \text{test\_result\_bin} == 0 \\ \text{I } & = & \text{length} \\ \end{array} $
.40	<pre>test_result_bin = Test_size- sum(test_result_array);</pre>	167	L = length( encriped_Alice_messege
41	sum (test_lesuit_allay),		);
.41		100	mas_bin = xor(transpose(
.42		168	Bob_Bits_same_base
.43	if test_result_bin == 0	169	$(\text{Test\_size} + 1:$
.45	disp ('Test Result – Eve	109	$Test\_size +L)$ ,
.40	as not detected')		encriped_Alice_messege
.46	app. TestresultTextArea.		);
. 10	Value = 'Test Result	170	str = char(bin2dec(
	- Eve as not detected		reshape (char (mas_bin+
	with confidence of:		,0,, 8,[]).,));
	$3 + \operatorname{string} ((1 - (3/4)^{\circ}))$	171	fprintf('Alice message
	Test_size) *100)+'%';	•	is: %s ', message);
.47	fprintf('Detecting Eve	172	fprintf('Bob get the
	confidance lavel - %s		message: %s ', str (1:
	precence, $(1-(3/4)^{})$		end));
	- ' ' ' '		* * *



.73	${\rm app.BobrecivedTextArea}.$	208	
	Value =	209	% Value changing function:
	convertCharsToStrings		$\operatorname{BobrecivedTextArea}$
	$(\operatorname{str});$ else	210	function  Pohyaciyad Tart Area Value Changing
.74	${ m app.Bobrecived Text Area}$ .		BobrecivedTextAreaValueChanging (app, event)
.75	Value = 'Simulation	211	changing Value = event . Value;
	as stoped do to a Eve		enunging value — event. value,
	detaction in test';	213	end
76	end	214	
.77	end	215	% Value changed function:
.78			ofbitsTextArea
.79	% Value changing function:	216	function
	of bits TextArea		of bits Text Area Value Changed  (
.80	function		$\operatorname{app}, \operatorname{event})$
	of bits Text Area Value Changing (	217	value = app.ofbitsTextArea.
	$\operatorname{app}, \operatorname{event})$		Value;
.81	changingValue = event.Value;	218	
.82	1	219	end
.83	end	220	07 V-111 f
.84	07 Value shanging function.	221	% Value changed function: MatchbasesAliceandBobTextArea
.85	% Value changing function: MatchbasesAliceandBobTextArea		function
0.0	function	222	${\it Matchbases Alice and Bob Text Area Value Cha}$
.86	${ m Matchbases Alice and Bob Text Area}$	ValueChang	
	(app, event)	223	value = app.
.87	changing Value = event. Value;	220	${ m Matchbases Alice and Bob Text Area}$
.88	enanging varae event varae,		. Value ;
.89	$\operatorname{end}$	224	, , , , , , , , , , , , , , , , , , , ,
.90		225	end
.91	% Value changing function:	226	
	matching bits with match bases Tex	t₂ <b>2√</b> rea	% Value changed function: matchingbitswithmatchbasesTextArea
.92	function		
	matchingbitswithmatchbasesText	tAsreaValue(	
	(app, event)		matching bits with match bases TextArea Value and the state of the s
.93	changingValue = event.Value;		(app, event)
.94	1	229	value = app.
.95	end		$ootnotesize{matchingbits} with match bases TextAre$
.96	07 17 1 1		. Value ;
.97	% Value changing function: ErrorTextArea	230	and
	function	231	end
.98	ErrorTextAreaValueChanging(	232	% Value changed function:
	app, event)	233	ErrorTextArea
.99	changing Value = event. Value;	234	function
100	onanging varao event. varae,	204	ErrorTextAreaValueChanged(app
201	$\operatorname{end}$		, event)
102		235	value = app.ErrorTextArea.
103	% Value changed function:		Value;
	EveEditField	236	
104	function	237	end
	EveEditFieldValueChanged (app,	238	
	event)	239	% Value changed function:
205	value = app.EveEditField.		${ m Testre sult Text Area}$
	$\operatorname{Value}$ ;	240	function
206	1		TestresultTextAreaValueChanged
207	$\operatorname{end}$		$(\mathrm{app},\ \mathrm{event})$



241	value = app.	273	% Create Panel
	TestresultTextArea. $Value$ ;	274	app.Panel = uipanel(app.
242			UIFigure);
243	$\operatorname{end}$	275	app. Panel. Position = [17 11
244	end		668 101];
245		276	1)
246	% Component initialization	277	% Create
247	methods (Access = private)	211	${ m Matchbases Alice and Bob Text Area Labe}$
	methods (necess = private)		With the about t
248	% Create UIFigure and components	0.00	ann
249		278	$\operatorname{app}. \\ \operatorname{MatchbasesAliceandBobTextAreaLabe}$
250	${\bf function} \ \ {\bf createComponents} \ ({\bf app})$		
251	07 0 4 41 6:1 41 6		= uilabel(app.Panel);
252	% Get the file path for	279	app.
	locating images		Matchbases Alice and Bob Text Area Labe
253	pathToMLAPP = fileparts (		. Horizontal Alignment = '
	mfilename('fullpath'));		right';
254		280	app.
255	% Create UIFigure and hide		${\bf Matchbases Alice and Bob Text Area Labe}$
	until all components are		. FontSize = 16;
	$\operatorname{created}$	281	app.
256	app.UIFigure = uifigure('		${\bf Matchbases Alice and Bob Text Area Labe}$
	Visible', 'off');		. Position = $[257\ 66\ 203]$
257	app. UIFigure. Position = [100]		22];
	100 983 685];	282	app.
258	app. UIFigure. Name = 'MATLAB	202	${ m Matchbases Alice and Bob Text Area Labe}$
200	App';		. Text = 'Match bases
0.50	TIPP ,		Alice and Bob';
259	% Crosto Imago		Affect and Dob ,
260	% Create Image	283	Of C
261	app.Image = uiimage (app.	284	% Create
	UIFigure);		${\bf Matchbases Alice and Bob Text Area}$
262	app.Image.Position = [340]		
	$121 \ 634 \ 490];$	285	app.
263	app.Image.ImageSource =		Matchbases Alice and Bob Text Area
	fullfile (pathToMLAPP,		= uitextarea(app.Panel);
	test help.png');	286	app.
264			${\bf Matchbases Alice and Bob Text Area}$
265	% Create		. Value Changed Fcn =
	${\bf Alice sendmes sage Button}$		createCallbackFcn (app,
266	app.AlicesendmessageButton =		@Matchbases Alice and Bob TextArea Value
	uibutton (app. UIFigure,		, true);
	state');	287	app.
267	app. AlicesendmessageButton.		${ m Matchbases Alice and Bob Text Area}$
	ValueChangedFcn =		. ValueChangingFcn =
	createCallbackFcn (app,		createCallbackFcn (app,
	@AlicesendmessageButtonVa	lueChanged	@MatchbasesAliceandBobTextAreaVal
	, true);	raconangea	, true);
	app. AlicesendmessageButton.		
268	Text = 'Alice send'	288	$\operatorname{app}. \\ \operatorname{MatchbasesAliceandBobTextArea}$
	message';		. FontSize = 16;
269	app. AlicesendmessageButton.	289	app.
	BackgroundColor = [0.3922]		${\bf Matchbases Alice and Bob Text Area}$
	$0.8314 \ 0.0745$ ;		. Position = $[475 \ 64 \ 182]$
270	${ m app.Alices}$ endmessage ${ m Button.}$		26];
	FontSize = 18;	290	
271	${\rm app.Alice sendmess age Button}.$	291	% Create of bits Text Area Label
	Position = $[74 \ 376 \ 183]$	292	app.ofbitsTextAreaLabel =
	36];		uilabel (app. Panel);
272	• •		, <del>-</del> - , , ,



93	<pre>app.ofbitsTextAreaLabel.    HorizontalAlignment = '    right';</pre>	. ValueChangedFcn = createCallbackFcn(app, @matchingbitswithmatchbasesTextArea
94	app.ofbitsTextAreaLabel.	, true);
95	A = A = A = A = A = A = A = A = A = A =	<pre>app.     matchingbitswithmatchbasesTextArea     .ValueChangingFcn =</pre>
96	app.ofbitsTextAreaLabel.Text = '# of bits';	$create Callback Fcn (app,\\ @matching bits with match bases Text Area of the control of the con$
97		, true);
98	% Create ofbitsTextArea 316	app.
99	<pre>app.ofbitsTextArea =     uitextarea(app.Panel);</pre>	<pre>matchingbitswithmatchbasesTextArea .FontSize = 16;</pre>
300	$rac{ m app.ofbitsTextArea}{ m ValueChangedFcn} =$	${ m app.} \\ { m matchingbitswithmatchbasesTextArea}$
	${ m createCallbackFcn}({ m app}, \\ { m @ofbitsTextAreaValueChanged}$	. Position = [267 19 114 26];
	, true);	17
101	app. ofbits Text Area.	% Create ErrorTextAreaLabel
	ValueChangingFcn = 320	app.ErrorTextAreaLabel =
	createCallbackFcn (app,	uilabel (app. Panel);
	@ofbitsTextAreaValueChanging	app. ErrorTextAreaLabel.
	, true);	HorizontalAlignment = '
102	app.ofbitsTextArea.FontSize	right';
	= 16; 322	$\operatorname{app}$ . $\operatorname{ErrorTextAreaLabel}$ .
103	$\operatorname{app.ofbitsTextArea}$ . Position	FontSize = 16;
304	$= [87 \ 65 \ 165 \ 25];$ 323	app.ErrorTextAreaLabel. Position = [414 21 61
105	% Create	22];
	$matching bits with match bases \hbox{\it $7$}\hbox{\it $4$} ext Area Label$	app.ErrorTextAreaLabel.Text = 'Error %';
106	$\operatorname{app}$ .	- Ellor // ,
.00	${\it matching bits with match bases} {\it Tsext Area Label}$	% Create ErrorTextArea
	= uilabel (app. Panel); 327	app. ErrorTextArea =
107	app.	uitextarea (app. Panel);
•	$matching bits with match bases \hbox{\it \$Ts} extArea Label$	app. ErrorTextArea.
	. Horizontal Alignment =	ValueChangedFcn =
	right';	createCallbackFcn (app,
108	app.	@ErrorTextAreaValueChanged
	matching bits with match bases TextArea Label	, true);
	. FontSize = 16;	app. Error Text Area.
109	app.	ValueChangingFcn =
	matching bits with match bases TextArea Label	${\tt createCallbackFcn}$ (app ,
	. Position = $[13 \ 21 \ 239 \ 22];$	<pre>@ErrorTextAreaValueChanging , true);</pre>
310	app. 330	app. ErrorTextArea. FontSize =
	matching bits with match bases TextArea Label	16;
	$.\mathrm{Text} = ^{,}\mathrm{matching}\mathrm{bits}_{331}$	$\operatorname{app}.\operatorname{ErrorTextArea}.\operatorname{Position} =$
	with match bases';	$[490 \ 20 \ 150 \ 25];$
11	332	
312	% Create	% Create
	matching bits with match bases TextArea	Teststringsize Edit Field Label
:13	app.  334	app.
	matchingbitswithmatchbasesTextArea	TeststringsizeEditFieldLabel
	= uitextarea (app. Panel);	= uilabel(app.UIFigure);
314	$^{335}$ matching bits with match bases ${ m TextArea}$	$\operatorname{app}$ . $\operatorname{TeststringsizeEditFieldLabel}$



	<pre>. HorizontalAlignment = ' right';</pre>	$create Callback Fcn (app,\\ @Alicemess age Edit Field Value Changed$
36	app. TeststringsizeEditFieldLabæl .FontSize = 16;	$ m , true); \ app. AlicemessageEditField. \ FontSize = 16; \$
37	app. $^{357}$ TeststringsizeEditFieldLabel . Position = [24 575 111	app. AlicemessageEditField. Position = [151 610 823 24];
	[22]; 358	% Create EveEditFieldLabel
38	app. 359	
	TeststringsizeEditFieldLab@l	app. EveEditFieldLabel =
	. Text = . Test string size	uilabel(app.UIFigure);
	;	app. EveEditFieldLabel.
39	W 0	HorizontalAlignment =
40	% Create	right';
	$Teststring size Edit Field \qquad {}_{362}$	$\operatorname{app}$ . $\operatorname{EveEditFieldLabel}$ .
41	${\tt app.TeststringsizeEditField}$	FontSize = 16;
	= uieditfield (app. 363	$\operatorname{app}$ . $\operatorname{EveEditFieldLabel}$ .
	UIFigure, 'numeric');	$Position = \begin{bmatrix} 348 & 645 & 70 \end{bmatrix}$
12	${ m app.Teststringsize Edit Field}$ .	23];
	ValueChangedFcn = 364	app.EveEditFieldLabel.Text =
	createCallbackFcn (app,	'Eve ';
	@TeststringsizeEditFieldValueChange	$\operatorname{ed}$
	, true);	% Create EveEditField
43	app. TeststringsizeEditField. 367	app. EveEditField =
	FontSize = $16$ ;	uieditfield (app. UIFigure,
14	app. TeststringsizeEditField.	'numeric');
	Position = $[143 \ 575 \ 122]$ 368	app. EveEditField.
	22];	ValueChangedFcn =
45	_ <b>_</b> ] ,	createCallbackFcn (app,
46	% Create	@EveEditFieldValueChanged
±0	AlicemessageEditFieldLabel	, true);
	369	app.EveEditField.FontSize =
		16;
47	${ m app}$ . ${ m AlicemessageEditFieldLabek}_{70}$	app.EveEditField.Position =
	= uilabel (app. UIFigure);	$[426 \ 646 \ 97 \ 22];$
40	, , ,	[420 040 31 22],
18	$^{371}$ Alicemess age Edit Field Labek $_{72}$	% Create
	. Horizontal Alignment = '	BobrecivedTextAreaLabel
	The state of the s	app. BobrecivedTextAreaLabel
	right';	
19	app.	= uilabel(app.UIFigure);
	AlicemessageEditFieldLabek <sub>74</sub>	app. BobrecivedTextAreaLabel.
	. FontSize = 18;	HorizontalAlignment = '
50	app.	right';
	AlicemessageEditFieldLabek <sub>75</sub>	app. BobrecivedTextAreaLabel.
	. Position = $[21 \ 611 \ 122]$	FontSize = 16;
	23]; 376	app. BobrecivedTextAreaLabel.
51	app.	$Position = \begin{bmatrix} 17 & 123 & 91 \end{bmatrix}$
	${\bf Alicemess age Edit Field Label}$	22];
	. Text = 'Alice message'; 377	$\operatorname{app}$ . Bobrecived $\operatorname{TextAreaLabel}$ .
52		Text = 'Bob recived';
53	% Create	
	${\rm Alicemessage Edit Field} \qquad {\rm _{379}}$	% Create BobrecivedTextArea
54	app.AlicemessageEditField = 380	$\operatorname{app}$ . Bobrecived $\operatorname{TextArea} =$
	uieditfield (app. UIFigure,	uitextarea (app. UIFigure);
	'text'); 381	$\operatorname{app}$ . Bobrecived $\operatorname{Text}\operatorname{Area}$ .
55	${ m app.Alicemess age Edit Field.}$	ValueChangingFcn =
	ValueChangedFcn =	createCallbackFcn (app,
	<del>-</del>	, ·



	$@Bobrecived Text Area Value {\bf Changing}\\$	
	, true);	FontName = Academy
382	$\operatorname{app}$ . Bobrecived $\operatorname{Text}\operatorname{Area}$ .	Engraved LET';
	FontSize = 16;	app.BB84SimulatorGUILabel.
383	$\operatorname{app}$ . Bobrecived $\operatorname{Text}\operatorname{Area}$ .	FontSize = 26;
	Position = $[122 \ 121 \ 849$	${ m app.BB84SimulatorGUILabel}$ .
	26];	FontWeight = 'bold';
384	409	${ m app.BB84SimulatorGUILabel}$ .
385	% Create	$FontColor = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix};$
	TestresultTextAreaLabel 410	${ m app.BB84SimulatorGUILabel}$ .
386	$\operatorname{app}.\operatorname{TestresultTextAreaLabel}$	$Position = \begin{bmatrix} 17 & 633 & 240 \end{bmatrix}$
	= uilabel(app.UIFigure);	35];
387	app. Testresult Text Area Label. 411	${ m app.BB84SimulatorGUILabel}$ .
	HorizontalAlignment = '	Text = 'BB84 Simulator
	right';	GUI';
388	app. TestresultTextAreaLabel. 412	
	FontSize = $16$ ;	% Show the figure after all
389	$\operatorname{app}$ . $\operatorname{Testresult}\operatorname{Text}\operatorname{AreaLabel}$ .	components are created
	Position = $[690 \ 88 \ 78$	app. UIFigure. Visible = 'on';
	22;	$\operatorname{end}$
390	app. TestresultTextAreaLabel. 416	end
	Text = 'Test result';	
391	418	% App creation and deletion
392	% Create TestresultTextArea 419	methods (Access = public)
	app. Testresult Text Area = 420	F 33311)
393	uitextarea (app. UIFigure); 421	% Construct app
204	app. TestresultTextArea. 422	function app = Sim_final
394	Value Changed Fcn = 423	
	G 111 1 D /	% Create UIFigure and
	${ m createCallbackFcn}$ (app , 424 ${ m @TestresultTextAreaValueChanged}$	components
		$\operatorname{createComponents}(\operatorname{app})$
	, true);	create components (app)
395	app. TestresultTextArea.	% Register the app with App
	FontSize = $16$ ; 427	Designer
396	app. TestresultTextArea.	
	Position = $[783 \ 11 \ 188 \ 428]$	registerApp (app, app.
	101];	$\mathrm{UIFigure})$
397	429	: f
398	% Create 430	$\inf \text{ nargout} = 0$
	fortrueOforfalseLabel 431	clear app
399	app.fortrueOforfalseLabel = 432	end
	uilabel (app. UIFigure); 433	$\operatorname{end}$
400	app.fortrueOforfalseLabel. 434	
	FontSize = 18;   435	% Code that executes before app
401	app.fortrue 0 for false Label.	deletion
	Position = $[530 \ 645 \ 163]$	${\rm function} \ \ {\rm delete}  ({\rm app})$
	[23];	Of D. L. HID.
402	app.fortrueOforfalseLabel. 438	% Delete UIFigure when app
	Text = '1 for true, 0 for	is deleted
	false';	delete (app. UIFigure)
403	440	end
404	% Create	end
	BB84SimulatorGUILabel 442 end	
405	app.BB84SimulatorGUILabel =	
	uilabel (app. UIFigure);	