



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials Bootcamp Style (Security 401)"
at <http://www.giac.org/registration/gsec>

An Introduction to Quantum Key Distribution

Derek Bastille
GSEC Practical Assignment Version 1.4
January 15, 2005

Abstract

Quantum Key Distribution (QKD) is a methodology for generating and distributing random one-time encryption keys using the principles of quantum physics. These keys are secure from eavesdropping or tampering while in transit from the sender to the receiver thanks to the fundamental principles of quantum physics which state that the process of observing a quantum system changes the system such that it is not possible to fully reconstruct its pre-measurement state.

QKD is only used for the transmission of keys and cannot be used as an encryption algorithm itself. However, it does provide a potential solution to the need to provide longer encryption keys to commonly used algorithms. Current research is focused on proving the security of QKD and increasing its range and stability. This paper provides a brief introduction to that research and outlines an example protocol and QKD implementation using the original Bennett and Brassard paper [2].

Overview

For as long as humans have been communicating, there has often been at least one extra participant in the conversation between the sender and the receiver—the eavesdropper. When the information being passed is intended to be public knowledge this is not a problem; however, when the message is intended for the receiver only, the participants need some way to keep their conversation private.

One way for the sender (who is typically referred to as Bob) to get a private message to Alice (the receiver) is to send the message in such a way that the eavesdropper (Eve) cannot get physical access to it while it is in transit between Bob and Alice. Unfortunately, this method is rarely practical and depends heavily on trusted third parties to get the message safely from Bob to Alice. The remaining method assumes that Eve has full access to the message while it is in route and, therefore, encrypts the message to prevent Eve from getting any information from it.

Many different algorithms have been developed over the years to satisfy this need to encrypt messages as they pass from sender to receiver. In the earliest ones, the security of the message depended entirely on the security of the algorithm itself. For example, the Caesarean (that is, attributed originally to Julius Caesar) cipher transposes the letters of a message by a specific number of places to encrypt it. Thus, the message: GET HELP might be transposed into JHW KHOS while it is in transit. This, obviously is not a very secure cipher since even if you do not know that the 'key' is shifting by 3 places, you can break it by trying all combinations of transpositions and/or analyzing the frequency of the letters [5].

One way to look at the effectiveness of an encryption method is to see to what degree that algorithm can transfer the randomness of the key to the message such that the message cannot be recovered without the key. For another example, here is a cipher that uses a key of the same length as the message that is to be encrypted. To see how it is used, let's go back to Bob and Alice.

In this case, Bob wants to send the message "Let's have lunch at Tony's" to Alice, but does not want to invite Eve. As shown in Figure 1, Bob first converts the letters in the message to numbers using a table known to both he and Alice.

	L	E	T	S	H	A	V	E	L	U	N	C	H	A	T	T	O	N	Y	S
	17	42	05	26	00	11	95	42	17	13	50	64	00	11	05	05	47	50	89	26
+	11	23	14	52	09	35	61	77	82	12	67	70	32	84	84	56	91	19	07	31
	28	65	19	78	09	46	56	19	99	25	17	34	32	95	89	51	38	69	86	57

Figure 1: A Vernam Cipher Example

Then, Bob uses a subset of a random string of digits that he has shared with Alice ahead of time. This set of random numbers is the same length as the message (Figure 1, row 2). Bob adds the random digits to the numbers derived from the message (ignoring carries) and sends the final message off to Alice.

Alice decodes the original message by subtracting the same subset of random digits from the encoded message and then converting the numbers back to letters. Once the message has been successfully transferred, both Bob and Alice destroy their copies of the random digits so that they cannot be reused. Since the encrypted message, while in transit, is essentially randomized by the key, Eve will be unable to decode it no matter how much knowledge she has of the algorithm or how much computing power she throws at it for a brute-force attack.

Encryption algorithms that use random keys of the same length as the message are known as Vernam ciphers (after Gilbert Vernam who invented them in 1918) [2] and have been proven to be unbreakable as long as the key is never reused for subsequent messages. One of the main disadvantages to Vernam ciphers is that the parties using it must find a way to securely exchange and store many sets of one-time keys for use in encrypting their messages. By extension, Vernam ciphers are also only useful between parties that already know and trust each other. Therefore, because of these impracticalities, Vernam ciphers are normally only used for highly sensitive military and diplomatic communications.

Many other ciphers and algorithms have been developed since the Vernam cipher, but nearly all have been broken by increasingly sophisticated mathematics and computer processing power. In fact, many would argue that it was the need to break advanced encryption schemes, such as Enigma, that pushed the development of electronic computers in the first place.

Throughout much of the 20th century there has been a sort of arms race between the developers of advanced encryption algorithms and the developers of advanced computing hardware that attempted to break those algorithms. Until the 1970s, however, all of the algorithms, machines and ciphers had the same basic problem as the Vernam cipher. Namely: How does Bob transfer his random key data to Alice

without having that data “leak” to Eve? Further: How does Alice store the key data in such a way that Eve cannot access it?

In 1976, Whitfield Diffie and Martin Hellman published a paper called “New Directions in Cryptography” [7] in which they presented the principles of Public Key Cryptography (PKC). In the original version of PKC, Bob and Alice each pick a pair of algorithms. The algorithms are complementary so that one can decrypt messages encrypted by the other. Then, they each make their chosen encryption algorithm public, but keep the decryption algorithm private. Bob and Alice can use these algorithms to send each other secret messages – without having to exchange secret keys in advance [3].

PKC was refined (and made practical) through research done by Ronald Rivest, Adi Shamir and Leonard Adleman at MIT in 1977 [2]. The Rivest-Shamir-Adleman (RSA) PKC algorithm neatly eliminated the need for Alice and Bob to share secret information prior to exchanging messages, but relies on the computational difficulty of figuring out the key from the encrypted message. Since the RSA cipher (as well as other PKC algorithms) uses a key that is shorter than the original message and is reused, it can be broken – given enough computational power and/or clever mathematics [GSEC Security Essentials, day 4].

The end result of all this is that the encryption arms race is forcing us to use longer and longer key lengths in both DES/AES type (private key) and RSA type (public/private key pair) algorithms to avoid being compromised by more powerful computer systems. Hence, one day we may find ourselves right back at the beginning again where Bob and Alice will need to share a fully random, one-time use key each time they want to exchange secret messages.

Again, we come to the question:

How do Bob and Alice share a secret key in such a way that Eve cannot intercept it while it is traveling from Bob to Alice?

Unfortunately, classical physics does not provide a way in which keys can be transported between Alice and Bob without the possibility of Eve intercepting it and, subsequently, decrypting Alice and Bob’s private communications. That is, no matter what method Alice uses to package the key for transport to Bob, there will always be some sort of sensor (or plain-old social engineering) that can read the key while it is in transition.

Quantum physics, on the other hand, does not hold such restrictions.

Ideal Quantum Key Distribution

Quantum physics is based on several theories of uncertainty. The most well known is the Heisenberg uncertainty principle which states that “The more precisely the position is known, the less precisely the momentum is known” [8]. In essence, the uncertainty principle means that the process of measuring one aspect of a quantum system changes that system and prevents you from knowing what the other aspects were prior to the measurement.

In about 1970, Stephen Wiesner wrote a paper entitled “Conjugate Coding” [6] in which he explored the possibility of using quantum physics to merge messages together so that you could read one or the other – but not both. Even though his paper was originally rejected (and remained unpublished for over 10 years), his idea of using

quantum physics was picked up by Charles Bennett and Gilles Brassard [2,9] and was eventually refined into a working prototype.

The original key distribution protocol used 2 orthogonal sets of photon polarizations¹ as the quantum basis for the distribution. That is, the polarization states in one set (horizontal or vertical), are distinct from those of any other set (clockwise or counter-clockwise). In their 1984 paper, Bennett, et al, used rectilinear and circular polarizations [9] for their sets, while in the later *Scientific American* article [2] the chosen polarizations were rectilinear and diagonal. For this paper, I have chosen rectilinear and diagonal polarizations for the examples.

Regardless of the polarizations chosen, the basic principle is the same: the detector for measuring one polarization set can only pass through photons that are polarized to a member of that set. Thus, photons that are, say, circularly polarized will be randomly re-polarized to either vertical or horizontal polarization if they pass through a rectilinear detector. For example, if Bob sends a photon that is polarized to 90° and Alice has her detector set to rectilinear, then the 90° polarization will be preserved. If, however, she has her detector set to measure diagonal polarizations, then the photon will be randomly re-polarized to either 45° or 135° and the original 90° polarization will be forever lost.

Similarly, if Eve attempts to measure any of Bob's photons while they are in transit and chooses the wrong polarization, then the original polarization will be lost when the photon passes through her detector. Thus, when Eve attempts to duplicate the photon for retransmission to Alice (since the original photon was consumed by the detector), she has a high probability of getting the polarization wrong. These wrong guesses will show up as errors in Alice's data.

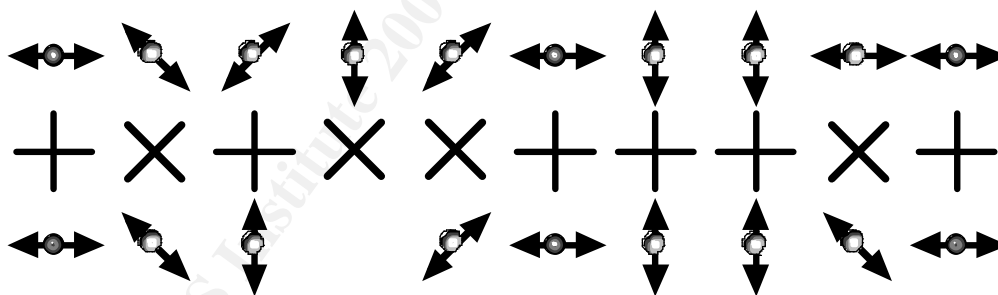






Figure 2: Example exchange between Bob and Alice (similar to illustration in [2])

Figure 2 shows the steps in an idealized exchange between Bob and Alice. The first step is for Bob to send a random series of photons with polarizations of 0° , 45° , 90°

¹ Photons are packets of energy that have both wave-like and particle-like behavior. Polarization refers to the angle of the photon's wave-like motion. Since photons are created with a random polarization, filters are used to choose photons with the specific polarization desired. For example, the lenses of Polaroid glasses have very thin lines engraved or printed on them. These lenses block photons that are vibrating perpendicular to the lines and, thus, filter out everything but a specific polarization. One interesting trick is to take 2 polarizing lenses and place one atop the other. Rotating the top lens 90° from the bottom lens will block any light – even though there are areas not covered by the lines on either lens [2].

or 135° (the top row of Figure 2). For each photon that she receives (not all of the photons that Bob generates will reach Alice), Alice randomly chooses to measure either rectilinear polarizations or diagonal polarizations (middle row). Alice jots down her measurement results but does not disclose them (bottom row).

Next, Alice publicly tells Bob which polarization set she used for each photon and Bob tells her which ones were correct. For this example, Alice received all of the photons sent by Bob, except for number 4, and picked the same polarization set for photons 1,2,5,6,7,8 and 10. Both Bob and Alice then discard all of the photons for which the polarization sets did not match (3 and 9 for this case).

Finally, the correct photons are converted to a binary sequence using a previously agreed upon method. For this example  and  are 0 while  and  are 1. For Bob and Alice, the end result is the string 0010110.

Bennette, et al, refer to the above steps as the *raw quantum transmission* [9] since no error correction or testing for eavesdropping has been done on the “raw” bits. In the ideal protocol, Bob would publicly disclose the actual polarizations for a random subset of the photons he sent to Alice (the photons in this subset would then be discarded since they would now be known to Eve). If there were no discrepancies in the subset, then Alice and Bob could reasonably assume that Eve did not attempt to eavesdrop on the transmission².

The remaining bits form the secret key that is shared between Bob and Alice. This key is used to encrypt/decrypt any messages exchanged between them using the Vernam cipher, Triple-DES, RSA or some other cryptographic algorithm.

Validation

In practice, there are two problems with the ideal approach to validating the raw quantum transmission: the first is that it is quite difficult to produce a one-photon pulse using current technology; the second is that photon detectors will always introduce some amount of random noise that will show up as errors in the measurements [8].

All of the actual implementations of QKD use a beam of very dim flashes of polarized light rather than individual photons. The average number of photons per flash is typically less than 1, but the possibility still exists for Eve to be able to split Bob's beam (so that part goes to her detector and the remainder goes to Alice). If the number of average photons per flash increases, then the chances of successful beam splitting also go up. Thus, Bob and Alice must always assume that Eve may know a subset of the raw quantum transmission.

² An important underlying assumption in the entire protocol is that Eve cannot impersonate either Bob or Alice for the public messages and that she cannot corrupt any of those messages. Otherwise, Eve could pretend to be Alice or Bob and wind up with a key that is shared with Bob and a key that is shared with Alice. Eve could then use those keys to decrypt/re-encrypt the secret messages and neither Bob nor Alice would know [9]. To counter this, Bob and Alice can authenticate each other by publicly exchanging a few bits from their shared key with each transmission. Since these sacrificial bits are then known to Eve, they are discarded.

One published scheme [9] to minimize the impact of errors (from either Eve or from random detector noise) is to do parity checking on the raw data. For this process, Bob and Alice first randomly shuffle the bits around to distribute any errors (Bob and Alice's copies of the key are shuffled identically to each other). They then split the key data into smaller blocks of size k – with k being small enough to minimize the chances of having more than one error in any one block. After splitting the key data, Bob and Alice compute and then compare the parity of each block. If the parity checks don't match, then the offending block is further split into sub-blocks and parity checked. The process is repeated until the error is found and corrected (by deleting the bad bit).

To avoid giving Eve any hints, Bob and Alice should drop one of the bits in each of the publicly discussed blocks (thus changing that block's parity). Once Bob and Alice have found all of the errors, they should randomly shuffle the bits in the key and then repeat the whole process. This will help find cases where there might have been an even number of errors in any one block. As a final check, they can pick a random subset of all of the remaining bits and do parity checking on it. By repeating this last check several times, Bob and Alice can be reasonably sure that they have a shared key that does not have any errors in it³[9].

One thing that they cannot be sure of, however, is the percent of this key that is known by Eve. To compensate for this, Bob and Alice use the principle of *privacy amplification*. This procedure calls for Bob and Alice to choose mathematical functions (such as hashing, parity, transposition, etc) to repeatedly transform the partially secret key into one that Eve know almost nothing at all about [2,9].

Other Protocols

The QKD protocol outlined here (often referred to as BB84 since it was first introduced by Bennette and Brassard in 1984 [2]) is certainly not the only one to have been developed thus far. A few of these other methods are:

1. Using only 2 polarization states [1: ref. 6]
2. Using the phase modulation of photons instead of polarization [1: ref. 7]
3. Using bright light [1, ref. 11]
4. Using Einstein-Podolsky-Rosen (EPR) entangled photon pairs [1, ref. 5]

In addition to being an alternative transmission protocol, EPR pairs also have the potential (theoretically [2]) to solve the problem of securely storing encryption keys.

It is beyond the scope of this paper to go into details about these other methods, but further information is available in the papers referenced both here and in the referenced bibliography.

³ In [2], Bennett, et al, dispenses with the idea of randomly shuffling and splitting the key into fixed blocks of size k and, instead, relies entirely on choosing random subsets of the data for parity checking. They further suggest that since randomly selected subsets have a 50% probability of detecting the error, than one could perform the same random block parity repeatedly to find all of the errors. With 20 rounds of testing, the probability of having undetected errors in the final key drops to $< .0001\%$.

Implementations

Several working prototypes for QKD have been developed. The first was developed using the BB84 protocol and was able to distribute keys over a free-space distance of 32cm.

As shown in Figure 3, the photon source for this system was dim flashes of light from a green LED. The light was focused through a lens and pinhole to provide a collimated beam and then was sent through a filter to horizontally polarize it. Pockels cells were used next to provide a way of rapidly changing the polarization from horizontal to vertical and to the two diagonal polarizations. At the other end of the 32cm air gap, the encoded beam was sent through another Pockels cell so that it could be rotated by 45° (to measure diagonal polarization), or not rotated (to measure rectilinear polarization). Once the beam had been through the receiving Pockels cell, it was sent through a calcite crystal, which split it and sent the vertically polarized photons to one detector and the horizontally polarized photons to the other [2]⁴. A PC that played the part of both Bob and Alice controlled each Pockels cell. The PC used a file of pre-generated random numbers to provide the random series of polarizations for the quantum transmission. Although it is not depicted in Figure 1, the prototype also had the capability of simulating Eve's activities [9].

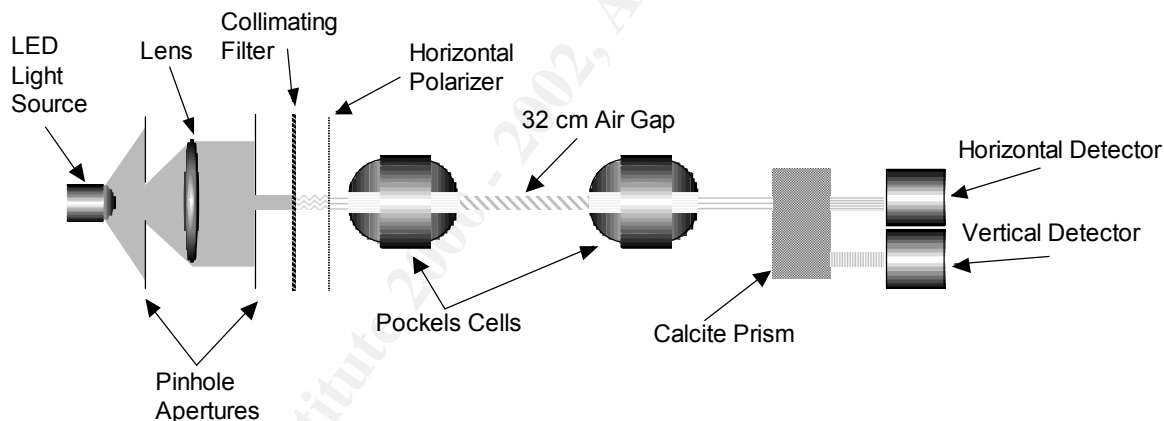


Figure 3: BB84 Prototype Quantum Key Distribution System (based on drawing in [2])

According to Bennett's 1991 paper [9], one of the runs of this system in February 1991 took about 10 minutes to perform the quantum transmission. In this run, Bob transmitted about 715,000 pulses of light and, of those, Alice correctly received about 2,000 pulses. After validation, error correction and privacy amplification, Bob and Alice wound up with a 754 bit long secret key.

When the run was repeated with a simulated Eve, Bob still correctly received 2,000 of the 715,000 pulses sent by Alice. Eve managed to intercept 336 bits of raw

⁴ A core component the original prototype was a prism made from calcite. This prism was (and still is in more recent prototypes) used to differentiate between horizontally and vertically polarized photons. The crystal prism will deflect photons that are vertically polarized, while horizontally polarized photons will pass straight through. The result is that the two rectilinear polarizations can be fed to separate detectors.

data, but also managed to raise Bob's error rate from 3.9% to 8%. The error correction, validation and privacy amplification resulted in 105 bits of shared data for Bob and Alice and an estimate of far less than 1 bit of data for Eve [9].

Since then, there have been many other prototype systems that have been implemented. While some of them have also been based on the BB84 protocol, a few have been based on the EPR model [1: ref 5] and others on the phase modulation protocol. In addition to trying out other protocols, these later prototypes have also been able to extend the range of quantum transmission to 10 kilometers (over optical fiber).

Yet another recent variation on the BB84 protocol [10] uses 4 separate lasers to generate the 4 polarization states (rather than a Pockels cell). The authors focused on increasing the quality and reliability of the transmission over long distances and managed to transfer 20 Kbit keys over 30 kilometers of optical fiber with very low error rates.

Practicalities

There are several issues that prevent QKD from being practical for general usage. These include:

- The distances are still limited to 10 to 30 kilometers
- It can take several minutes to transfer and validate just one key
- The current systems are large, bulky, delicate and very expensive
- Two systems cannot share the same optical fiber at the same time

Also, none of the protocols deal with the problem of authentication. That is, there is no way for Bob to know that he is really talking to Alice and vice-versa. A technique called oblivious transfer could solve this last problem, but it is rather unwieldy and can consume thousands of photons each time Bob and Alice try to use it [3].

Distribution Vs Cryptography

The term Quantum Cryptography in this context is somewhat of a misnomer since none of these systems or protocols are useful for actually doing cryptography. Thus, we will still need to rely on current well-known cryptographic algorithms such as AES and Triple-DES to do the encryption/decryption of messages. This is not to say, however, that quantum mechanics is not useful for cryptography—just that the label *Quantum Key Distribution* is probably the more accurate term.

The use of quantum computers in large number factorization and/or code breaking is discussed in the *Physics Today* article [11]. Dr. Gottesman proposes the possibility of a somewhat bleak future for public key cryptography where quantum computers have rendered them obsolete. However, he also thinks that QKD might help to provide some security through private key, Vernam-type ciphers.

Conclusion

The art and science of cryptography has changed tremendously over the last several thousand years from simple transposition algorithms to today's highly complex mathematical ciphers. Even with all of these changes, however, the basic idea has remained the same: Bob and Alice need to share a bit of secret information between them that they can use to obfuscate messages that they exchange over public

channels. This obfuscation is needed whenever Bob and Alice need to prevent Eve from reading any messages that she has managed to intercept.

Eve's tools for breaking the algorithms used by Bob and Alice, of course, have also changed over the years. These computational tools allow Eve to break apart ever more complex ciphers and force Bob and Alice to use longer and longer randomized keys; keys that also need to be refreshed much more frequently.

The use of quantum transmission to distribute cryptographic keys is an elegant solution to this problem of needing longer and more frequently generated keys to deal with Eve's increased computational capabilities. These transmissions rely on the fundamental quantum physics principle of uncertainty to make it incredibly difficult for Eve to intercept enough of the exchanged keys to be able to use them to decrypt Bob and Alice's messages.

Current QKD systems are still experimental and are not yet practical for general usage. However, the field is advancing rapidly and today's prototypes are able to generate and transfer longer keys over distances of 10 to 30 kilometers or more. Proofs have also been published recently that validate the absolute security of QKD methods [12]. Even with these advances, however, and the absolute security provided by QKD, it is still only useful for transferring keys between two parties who already know each other. That is, if Alice and Bob do not already know (or trust) each other, then methods other than QKD must be used to establish that trust.

Lastly, quantum mechanics itself is a double-edged sword. For all of the security benefits of using QKD to exchange keys, the use of quantum computers also holds the potential to break all known public key ciphers (and most private key ciphers as well). "Only time will tell who benefits more from quantum cryptography: the code-makers or the code-breakers." [Dr. Gottesman in 11]

Bibliography

There are quite a few sources on the web these days related to the fields of Quantum key Distribution and Quantum Cryptography. A comprehensive bibliography covering many of these papers was written by Gilles Brassard of the Université de Montréal. The most recent update of this Bibliography was in 1998:

1. Brassard, G., and Crépeau, C, "A Bibliography of Quantum Cryptography", <http://www.cs.mcgill.ca/~crepeau/CRYPTO/Biblio-QC.html>

The main source for this paper was an article in Scientific American published in 1992:

2. Bennett, C.H., Brassard, G. and Ekert, A.K., "Quantum Cryptography", *Scientific American*, October 1992, pp. 50 – 57. A scanned copy is available at: <http://www.dhshara.com/book/quantcos/aq/qcrypt.htm#anchor618596>

Other sources (see: <http://www.google.com> "Quantum Key Distribution")

3. Bennett, C.H., "Quantum Cryptography: Uncertainty in the Service of Privacy", *Science*, Vol. 257, 7 August 1992, pp. 752 – 753.
4. Barnette, C.H., and Phoenix, S.J.D., "The Principles of Quantum Cryptography", *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*,

- Vol. 354, Issue 1708, Nonlinear Optics for Information Processing and Communication (15 March, 1996), pp. 793 – 803.
5. Wiesner, S., “Conjugate coding”, *Sigact News*, vol. 15.no. 1, 1983, pp. 78 – 88.
 6. Diffie, W., Hellman, M.E. “New Directions in Cryptography”, *IEEE Transactions on Information Theory* **IT-22**, 1976, pp. 644 – 654.
 7. American Institute of Physics, Heisenberg exhibit:
<http://www.aip.org/history/heisenberg/>
 8. Bennette, C.H., Bessette, F., Brassard, G., Salvail, L. and Smolin, J., “Experimental Quantum Cryptography”, *Journal of Cryptology*, vol. 5, no. 1, 1992, pp. 3 – 28. Preliminary version in *Advances in Cryptology – Eurocrypt '90 Proceedings*, May 1990, Springer – Verlag, pp. 253 – 265.
 9. Flam, F., “Quantum Cryptography’s Only Certainty: Secrecy”, *Science*, vol. 253, 1991, pp. 858.
 10. Zbinden, H., Gisin, N., Huttner, B., Muller, A. and Tittel, W., “Practical Aspects Quantum Cryptographic Key Distribution”, *Journal of Cryptology*, vol. 13, 2000, pp. 207 – 220.
 11. Gottesman, D. and Lo, H.K., “From Quantum Cheating to Quantum Security”, *Physics Today*, November 2000. Available online:
<http://www.aip.org/pt/vol-53/iss-11/captions/p22cap3.html/>
 12. Lo, H.K., “A Simple Proof of the Unconditional Security of Quantum Key Distribution”, Hp Laboratories Bristol, May 1999. Available online:
<http://www.hpl.hp.com/techreports/1999/HPL-1999-63.pdf>
 13. Biham, E., Boyer, M., Brassard, G., van de Graaf, J. and Mor, T., “Security of Quantum Key Distribution Against All Collective Attacks”, December, 2001. Available online: http://www.iro.umontreal.ca/~boyer/BBBGM_010808n.pdf

© SANS Institute 2000

Upcoming Training

Click Here to
{Get CERTIFIED!}



SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
Community SANS Indianapolis SEC401	Indianapolis, IN	Oct 09, 2017 - Oct 14, 2017	Community SANS