A Propositional Cryptanalysis of Pfleeger & Pfleeger's Claim of Solution to the Problem of the Person in the Middle, via the Schema $E(k_{PUB-B}, E(k_{PRI-A}, K))$ [1]

Manuel A Carro, *MSc. CPEN UT Candidate, SUAGM & Principal, Atelier Limited*

***Abstract*—In order to preserve the integrity of secrecy systems [2] throughput transmissions across the channel, confidential key distribution ought beseech primordia. Public key cryptography prognoses an intrinsic indisputable alternative; however, as temporal accessibility become infinitesimal deliverables of transmission, so do costs in processing power. As such, it's desirable to exercise asymmetric schemata insofar distribution of symmetric keys, so that parties amidst communications could arbitrage against latency costs consequential of the secrecy. Pfleeger & Pfleeger in [1] exposé a scheme for the asymmetric distribution of symmetric keys, claimant of being immune to the problem of the person in the middle or man-in-the-middle attack. In this work we propose that enunciating such a schema is least sufficiently flawed, and utter mathematical argument and python simulation to show that under certain conditions, said distribution exertion is vulnerable to a man-in-the-middle attack.**

***Keywords*—Public key cryptography, key distribution, man-in-the-middle attack.**

I. Mathematical Formulae

In this section, we expose the mathematical formulation underlying the man-in-the-middle attack against Pfleeger & Pfleeger's asymmetric symmetric-key distribution schema, $E(k_{PUB-B}, E(k_{PRI-A}, K))$. We begin by enunciating the fundamental equations of public key cryptography as articulated by Stallings [3], and proceed to outline the general sequence and equations of the attack, in the systems' sense. For simplicity, we define the following nomenclature:

- K: Symmetric key
- $\hat{K}$: Compromised symmetric key
- $KAPRI$: Alice's private key
- $KAPUB$: Alice's public key
- $KBPRI$: Bob's private key
- $KBPUB$: Bob's public key
- $KAMPRI$: Malvo's Alice private key
- $KAMPUB$: Malvo's Alice public key
- $KBMPRI$: Malvo's Bob private key
- $KBMPUB$: Malvo's Bob public key

A. General Asymmetric Equations

If Alice and Bob are agents intending to communicate via public key cryptography across a channel, there are two possible schemes that could be exercised:

Scheme 1: Alice publishes her public key, $KAPUB$, and encrypts the plaintext with her private key, $KAPRI$. Bob then proceeds to decrypt the ciphertext using Alice's public key, $KAPUB$:

$$C = E(KAPRI, P) \quad P = D(KAPUB, C)$$

Note that

$$P = D(KAPUB, E(KAPRI, P))$$

Scheme 2: Bob publishes his public key, $KBPUB$, and Alice encrypts the plaintext using Bob's public key, $KBPUB$. Bob then proceeds to decrypt the ciphertext using his private key, $KBPRI$:

$$C = E(KBPUB, P) \quad P = D(KBPRI, C)$$

Similarly,

$$P = D(KBPRI, E(KBPUB, P))$$

B. Sequence of Attack

Imagine that Alice desires to distribute Bob a symmetric key to secure communications across an interceptable channel using a symmetric cipher, such as the Advanced Encryption Standard (AES). Pfleeger & Pfleeger show that asymmetrically transmitting the key using either of the above schemes is susceptible to a man-in-the-middle [1]. Instead, they propose the schema $E(k_{PUB-B}, E(k_{PRI-A}, K))$ to to mitigate against the person in the middle, but fail to observe that it in itself is also vulnerable to a man-in-the-middle attack.

If Malvo is the person in the middle, and assuming that he knows the symmetric/asymmetric ciphers being used by Alice and Bob, then the attack might unravel as follows:

1) Alice sends her public key to Bob, and Bob sends his public key to Alice.
2) Malvo intercepts and stores both public keys in his repository, and:

M. Carro is a graduate student with the Department of Electrical and Computer Engineering of Universidad del Turabo, SUAGM.

Manuscript submitted December 15, 2016.

a) Impersonating Alice sends Bob a new public hey of his own, $KAMPUB$;
b) Impersonating Bob sends Alice another anew public key of his own, $KBMPUB$.

3) Alice enciphers the symmetric key, K, using Pfleeger & Pfleeger's schema, however unsuspectingly using Malvo's Bob public key $KBMPUB$ instead of Bob's actual public key, $KBPUB$, and sends the ciphertext to Bob.
4) Malvo intercepts Alice's ciphertext to Bob and decrypts it using Alice's public key, $KAPUB$, and Malvo's Bob private key, $KBMPRI$, thus extracting the symmetric key.
5) Malvo generates a compromised public key, $\hat{K}$, and encrypts it using Pfleeger & Pfleeger's schema with Malvo's Alice private key, $KAMPRI$, and Bob's public key, $KBPUB$, and sends the ciphertext to Bob by impersonating Alice.
6) Bob decrypts the compromised ciphertext by using Malvo's Alice public key, $KAMPUB$, and Bob's private key, $KBPRI$, to recover the compromised symmetric key, $\hat{K}$.

Alice and Bob now think that they both share the same symmetric key; however evidently, this is not the case. Subsequently,

If Alice sends a message to Bob:

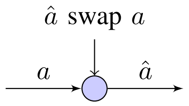
1) Alice will symmetrically encrypt the plaintext using the symmetric key, K, and send the ciphertext to Bob.
2) Malvo will intercept Alice's ciphertext, decrypting it using his extracted symmetric key, K.
3) Malvo reads the plaintext and re-encrypts it using the compromised symmetric key, $\hat{K}$, and sends the compromised ciphertext to Bob by impersonating Alice.
4) Bob decrypts the compromised ciphertext by using the compromised symmetric key, $\hat{K}$, to recover the plaintext.

If Bob sends a message to Alice:

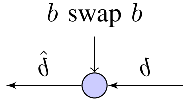
1) Bob will symmetrically encrypt the plaintext using the compromised symmetric key, $\hat{K}$, and send the compromised ciphertext to Alice.
2) Malvo will intercept Bob's compromised ciphertext, decrypting it using his compromised symmetric key, $\hat{K}$.
3) Malvo reads the plaintext and re-encrypts it using the symmetric key, K, and sends the ciphertext to Alice by impersonating Bob.
4) Alice decrypts the ciphertext by using the symmetric key, K, to recover the plaintext.

C. *Systems Dynamics of Attack*

Malvo intercepts and stores both public keys in his repository, and swaps Alice's public key, $KAPUB$ with Malvo's Alice public key, $KAMPUB$, and Bob's public key, $KBPUB$, with Malvo's Bob public key, $KBAPUB$:

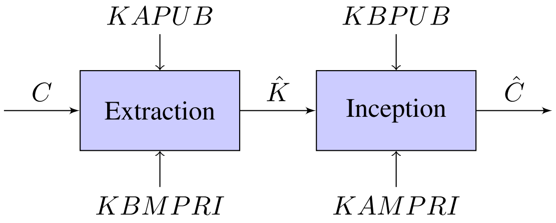


$$a = KAPUB,\ \hat{a} \text{ swap } a = \hat{a},\ \hat{a} = KAMPUB$$



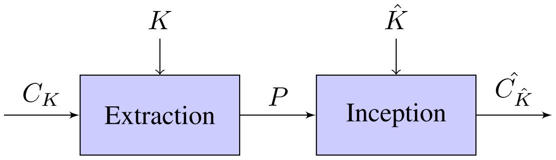
$$b = KBPUB,\ \hat{b} \text{ swap } b = \hat{b},\ \hat{b} = KBMPUB$$

Malvo intercepts ciphertext from Alice and extracts symmetric key, generates compromised symmetric key, then impersonates Alice to send compromised ciphertext to Bob:



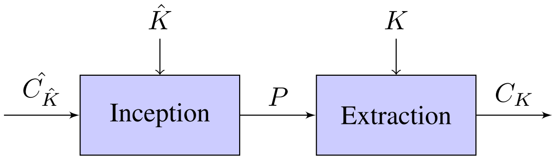
$$C = E(KBMPUB, E(KAPRI, K))$$
$$\hat{K} = T\{D(KBMPRI, D(KAPUB, K))\}$$
$$\hat{C} = E(KBPUB, E(KAMPRI, \hat{K}))$$

Alice and Bob think they both share the same symmetric key; Alice symmetrically encrypts a message with key K, Malvo intercepts it, extracts the plaintext, then re-encrypts it using the compromised symmetric key $\hat{K}$, and sends Bob compromised ciphertext by impersonating Alice:



$$C_K = E(P, K),\ P = D(C_K, K),\ \hat{C_{\hat{K}}} = E(P, \hat{K})$$

Conversing the latter:



$$\hat{C_{\hat{K}}} = E(P, \hat{K}),\ P = D(\hat{C_{\hat{K}}}, \hat{K}),\ C_K = E(P, K)$$

II. Simulacrum Algorithmica

In order to computationally demonstrate the vulnerability of Pfleeger & Pfleeger's schema to a man-in-the-middle attack, a scripted simulation was written in python27 fundamentally as a function of the pycrypto 2.6.1 package. The scripted simulation was remotely assembled and executed thereof on a virtual private server (VPS) operating the Free Berkeley Software Distribution (FreeBSD) 10.3-RELEASE. For the asymmetric schema, the Rivest-Shamir-Adleman (RSA) cryptosystem with key sizes of 1024 and 2048 was used according to the encryption protocol PKCS#1 OAEP. The reason for the use of two different key sizes over RSA was technically due to the observation that the python `Crypto.Cipher.PKCS1_OAEP` module required the "internal" key to be lesser in size than the "external" key for dual RSA encryption. The nature of this requirement, insofar theoretical or computational, was not explored. For the symmetric cipher, the Advanced Encryption Standard via the module `Crypto.Cipher.AES` was used on base 64 (8 bytes) and a key size of 128 bits (192 or 256 also permissible).

Asymmetric key generation was attained via the use of OpenSSH, invoked from the Bourne-again shell (Bash shell) on FreeBSD:

```
# ssh-keygen -b key_size -t RSA
```

A tuple of eight RSA key-pairs was spawn, for a total of sixteen encompassing keys. The project's full repository is available online and cloneable through GitHub by executing on shell:

```
# git clone https://github.com/praxepraxis
~/CPEN503.git
```

In the latter, the prefix ~ in ~/CPEN503.git is meant to symbolize a partition of sentence resultant from the manuscript's column width. Tildes before new hashtags should be interpreted as concatenation across the hashtag's instance, and ignored upon exertion in code.

The main, `mills_malvo.py`, executed successfully in FreeBSD 10.3 and Windows 8.1 64-bit operating systems. It's callable by exercising:

```
# python mills_malvo.py
```

The program will prompt the user for a series of inputs throughout. For outputs hereafter inputs were:

```
K = Sixteen byte key

K_hat = Sixteen bite key

P = Nos vemos en la riviera te llamo de
    Monaco, zarpando de Malta
```

Output from the BSD system:

```
    $ pwd
/usr/home/funkkagalaxxia/CPEN/FinalProject
    /CPEN503
$ sudo python mills_malvo.py
Password:
Monserrate-Mills-Malvo 1.7 Attack 0.1 CPEN
     503 Final Project Crypto
Copyright 2016 Atelier-Velvet Corporation.

ATTACK ON THE ``SECURE'' SCHEMA POSTULATED
     BY THE BOOK: RSA[KBPUB, RSA[KAPRI, K
    ]]--->ARCRSA[KBPRI, ARCRSA[KAPUB, RSA[
    KBPUB, RSA[KAPRI, K]]]]:

Order of Events:

INTERCEPTION OF THE PUBLIC KEYS:

[TRANSMISSION FROM ALICE INTENDED TO BOB]:
     Alice sends Bob her public key, KAPUB
    .
[TRANSMISSION FROM ALICE TO BOB
    INTERCEPTED BY MALVO]: Malvo swaps
    Alice's␣public␣key,␣KAPUB,␣with␣a␣
    public␣key␣from␣the␣pairs␣of␣his␣own,␣
    KAMPUB.
[TRANSMISSION␣FROM␣MALVO␣TO␣BOB]:␣Malvo␣
    sends␣Bob␣the␣swap␣of␣Alice's public
    key, KAMPUB, and stores Alice's␣public
    ␣key,␣KAPUB,␣in␣his␣key␣repository.
[TRANSMISSION␣FROM␣BOB␣INTENDED␣TO␣ALICE]:
    ␣Bob␣sends␣Alice␣his␣public␣key,␣KBPUB
    .
[TRANSMISSION␣FROM␣BOB␣TO␣ALICE␣
    INTERCEPTED␣BY␣MALVO]:␣Malvo␣swaps␣Bob
    's public key, KBPUB, with another
    public key from the pairs of  his own,
     KBMPUB.
[TRANSMISSION FROM MALVO TO ALICE]: Malvo
    sends Alice the swap of Bob's␣public␣
    key,␣KBMPUB,␣and␣stores␣Bob's public
    key, KBPUB, in his key repository.

EXTRACTION OF THE SYMMETRIC KEY:

[GENERATION OF THE 16 BYTE SYMMETRIC KEY
    BY ALICE]:
Alice, enter the symmetric key and press
    enter to send to Bob: sixteen byte key
Allice entered:  sixteen byte key

[DOUBLE RSA ENCRYPTION AND TRANSMITTAL OF
    SYMMETRIC KEY USING ALICE'S␣PRIVATE␣
    KEY,␣THEN␣MALVO'S BOB COMPROMISED
    PUBLIC KEY]:

[MALVO'S␣DOUBLE␣RSA␣DECRYPTION␣OF␣ALICE'S
    CIPHERTEXT TO BOB USING ALICE'S␣PUBLIC
    ␣KEY␣AND␣MALVO'S BOB PRIVATE KEY,
    KBMPRI]:
[MALVO'S␣EXTRACTED␣SYMMETRIC␣KEY␣PAR␣
```

```
   INTERCEPTION␣FROM␣ALICE␣TO␣BOB␣IS]:␣␣
   sixteen␣byte␣key

[GENERATION␣OF␣COMPROMISED␣SYMMETRIC␣KEY␣
   FOR␣MALVO␣TO␣SEND␣BOB␣AS␣IF␣WERE␣
   COMMING␣FROM␣ALICE]
Enter␣the␣compromised␣symmetric␣key␣to␣
   send␣to␣Bob␣in␣the␣name␣of␣Alice,␣
   Malvoo␣....␣:␣sixteen␣bite␣key
Malvo␣entered:␣␣sixteen␣bite␣key

[DOUBLE␣RSA␣ENCRYPTION␣AND␣TRANSMITTAL␣OF␣
   COMPROMISED␣SYMMETRIC␣KEY␣USING␣MALVO'
   S ALICE PRIVATE KEY, THEN BOB'S␣PUBLIC
   ␣KEY]:

[BOB'S DOUBLE RSA DECRYPTION OF MALVO'S␣
   COMPROMISED␣CIPHERTEXT␣USING␣BOB'S
   PRIVATE KEY AND MALVO'S␣ALICE␣PUBLIC␣
   KEY,␣KAMPUB]:
[BOB␣RECEIVES␣COMPROMISED␣SYMMETRIC␣KEY]:␣
   ␣sixteen␣bite␣key

[ALICE␣AND␣BOB␣NOW␣THINK␣THEY␣SHARE␣THE␣
   SAME␣SYMMETRIC␣KEY␣...␣MALVO␣KNOWS␣
   THEY'LL BE USING AES TO TRANSMIT
   ACROSS THE CHANNEL ....]

[ALICE ---> BOB]: ALICE AES ENCRYPTS
   MESSAGE MA ON THE BLOCKSIZE (16 BYTES)
    WITH KEY K (16, 24, OR 32 BITES) AND
   SENDS IT TO BOB

Alice, enter your message to bob ... It's␣
   secure␣!:␣Nos␣vemos␣en␣la␣riviera␣te␣
   llamo␣de␣Monaco,␣zarpando␣de␣Malta
Alice's message MA was:  Nos vemos en la
   riviera te llamo de Monaco, zarpando
   de Malta

[MALVO INTERCEPTS ALICE'S␣AES␣ENCRYPTED␣
   CIPHERTEXT␣TO␣BOB␣AND␣DECRYPS␣IT␣USING
   ␣ALICE'S EXTRACTED SYMMETRIC KEY]
[MALVO RECOVERS AND READS ALICE'S␣MESSAGE␣
   TO␣BOB␣...]:␣␣  f N  k <'>
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
   ␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣  ˜    N o s ␣
   vemos␣en␣la␣riviera␣te␣llamo␣de␣Monaco
   ,␣zarpando␣de␣Malta

[MALVO␣NOW␣RE␣AES␣ENCRYPTS␣ALICE'S MESSAGE
    TO BOB BUT USING THE COMPROMISED
   SYMMETRIC KEY, K HAT]:

[BOB DECRYPTS THE COMPROMISED AES
   CIPHERTEXT USING THE COMPROMISED
   SYMMETRIC KEY, K HAT]:
Here it is Bob, the message so securely
   sent by Alice ;):      : f 8 m    W W
   [.  J v  f N  k <'>
Nos vemos en la riviera te llamo de Monaco
   , zarpando de Malta
```

Output from the Windows system:

```
 ]Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights
    reserved.

C:\Users\stconh>cd c:\Users\stconh\
   Documents\FinalProjectCrypto\CPEN503

c:\Users\stconh\Documents\
   FinalProjectCrypto\CPEN503>dir
 Volume in drive C is TI10684500B
 Volume Serial Number is E0F2-44FC

 Directory of c:\Users\stconh\Documents\
    FinalProjectCrypto\CPEN503

12/13/2016  09:42 PM    <DIR>          .
12/13/2016  09:42 PM    <DIR>          ..
12/13/2016  09:42 PM    <DIR>
   CPEN503
12/13/2016  09:27 PM             1,702 KA.
   der
12/13/2016  09:27 PM               398 KA.
   der.pub
12/13/2016  09:27 PM             1,706
   KAattack.der
12/13/2016  09:27 PM               398
   KAattack.der.pub
12/13/2016  09:27 PM               902 KB.
   der
12/13/2016  08:26 PM               229 KB.
   der.pub
12/13/2016  09:27 PM               902
   KBattack.der
12/13/2016  09:27 PM               230
   KBattack.der.pub
12/13/2016  09:27 PM             6,737
   mills_malvo.py
12/13/2016  09:27 PM                11
   README.md
              10 File(s)         13,215
                 bytes
               3 Dir(s)  13,623,750,656
                  bytes free

c:\Users\stconh\Documents\
   FinalProjectCrypto\CPEN503>python
   mills_malvo.py
Monserrate-Mills-Malvo 1.7 Attack 0.1 CPEN
    503 Final Project Crypto
Copyright 2016 Atelier-Velvet Corporation.
```

```
ATTACK ON THE ‘‘SECURE’’ SCHEMA POSTULATED
     BY THE BOOK: RSA[KBPUB, RSA[KAPRI, K]
]--->ARCRSA[KBPRI, ARCRSA[KAPUB, RSA[KBPUB
    , RSA[KAPRI, K]]]]:

Order of Events:

INTERCEPTION OF THE PUBLIC KEYS:

[TRANSMISSION FROM ALICE INTENDED TO BOB]:
     Alice sends Bob her public key, KAPUB
.
[TRANSMISSION FROM ALICE TO BOB
    INTERCEPTED BY MALVO]: Malvo swaps
    Alice’s␣publi
c␣key,␣KAPUB,␣with␣a␣public␣key␣from␣the␣
    pairs␣of␣his␣own,␣KAMPUB.
[TRANSMISSION␣FROM␣MALVO␣TO␣BOB]:␣Malvo␣
    sends␣Bob␣the␣swap␣of␣Alice’s public
    key
, KAMPUB, and stores Alice’s␣public␣key,␣
    KAPUB,␣in␣his␣key␣repository.
[TRANSMISSION␣FROM␣BOB␣INTENDED␣TO␣ALICE]:
    ␣Bob␣sends␣Alice␣his␣public␣key,␣KBPUB
.
[TRANSMISSION␣FROM␣BOB␣TO␣ALICE␣
    INTERCEPTED␣BY␣MALVO]:␣Malvo␣swaps␣Bob
    ’s public
key, KBPUB, with another public key from
    the pairs of  his own, KBMPUB.
[TRANSMISSION FROM MALVO TO ALICE]: Malvo
    sends Alice the swap of Bob’s␣public␣k
ey,␣KBMPUB,␣and␣stores␣Bob’s public key,
    KBPUB, in his key repository.

EXTRACTION OF THE SYMMETRIC KEY:

[GENERATION OF THE 16 BYTE SYMMETRIC KEY
    BY ALICE]:
Alice, enter the symmetric key and press
    enter to send to Bob: sixteen byte key
Allice entered:  sixteen byte key

[DOUBLE RSA ENCRYPTION AND TRANSMITTAL OF
    SYMMETRIC KEY USING ALICE’S␣PRIVATE␣KE
Y,␣THEN␣MALVO’S BOB COMPROMISED PUBLIC KEY
    ]:

[MALVO’S␣DOUBLE␣RSA␣DECRYPTION␣OF␣ALICE’S
    CIPHERTEXT TO BOB USING ALICE’S␣PUBLIC
␣KEY␣AND␣MALVO’S BOB PRIVATE KEY, KBMPRI]:
[MALVO’S␣EXTRACTED␣SYMMETRIC␣KEY␣PAR␣
    INTERCEPTION␣FROM␣ALICE␣TO␣BOB␣IS]:␣␣
    sixtee
n␣byte␣key

[GENERATION␣OF␣COMPROMISED␣SYMMETRIC␣KEY␣
    FOR␣MALVO␣TO␣SEND␣BOB␣AS␣IF␣WERE␣
    COMMIN
G␣FROM␣ALICE]
Enter␣the␣compromised␣symmetric␣key␣to␣
    send␣to␣Bob␣in␣the␣name␣of␣Alice,␣
    Malvoo
....␣:␣sixteen␣bite␣key
Malvo␣entered:␣␣sixteen␣bite␣key

[DOUBLE␣RSA␣ENCRYPTION␣AND␣TRANSMITTAL␣OF␣
    COMPROMISED␣SYMMETRIC␣KEY␣USING␣MALVO’
S ALICE PRIVATE KEY, THEN BOB’S␣PUBLIC␣KEY
    ]:

[BOB’S DOUBLE RSA DECRYPTION OF MALVO’S␣
    COMPROMISED␣CIPHERTEXT␣USING␣BOB’S
    PRIVA
TE KEY AND MALVO’S␣ALICE␣PUBLIC␣KEY,␣
    KAMPUB]:
[BOB␣RECEIVES␣COMPROMISED␣SYMMETRIC␣KEY]:␣
    ␣sixteen␣bite␣key

[ALICE␣AND␣BOB␣NOW␣THINK␣THEY␣SHARE␣THE␣
    SAME␣SYMMETRIC␣KEY␣...␣MALVO␣KNOWS␣
    THEY’
LL BE USING AES TO TRANSMIT ACROSS THE
    CHANNEL ....]

[ALICE ---> BOB]: ALICE AES ENCRYPTS
    MESSAGE MA ON THE BLOCKSIZE (16 BYTES)
     WITH
 KEY K (16, 24, OR 32 BITES) AND SENDS IT
     TO BOB

Alice, enter your message to bob ... It’s␣
    secure␣!:␣Nos␣vemos␣en␣la␣riviera␣te␣l
lamo␣de␣Monaco,␣zarpando␣de␣Malta␣...
Alice’s message MA was:  Nos vemos en la
    riviera te llamo de Monaco, zarpando
    de
 Malta ...

[MALVO INTERCEPTS ALICE’S␣AES␣ENCRYPTED␣
    CIPHERTEXT␣TO␣BOB␣AND␣DECRYPS␣IT␣USING
    ␣A
LICE’S EXTRACTED SYMMETRIC KEY]
[MALVO RECOVERS AND READS ALICE’S␣MESSAGE␣
    TO␣BOB␣...]:␣␣  2   ]}84  @ ˜41& B2Nos
    ␣vemo
s␣en␣la␣riviera␣te␣llamo␣de␣Monaco,␣
    zarpando␣de␣Malta␣...

[MALVO␣NOW␣RE␣AES␣ENCRYPTS␣ALICE’S MESSAGE
     TO BOB BUT USING THE COMPROMISED SYMM
ETRIC KEY, K HAT]:

[BOB DECRYPTS THE COMPROMISED AES
    CIPHERTEXT USING THE COMPROMISED
    SYMMETRIC KEY
```

```
, K HAT]:
Here it is Bob, the message so securely
   sent by Alice ;):                 g
   +           Q   2   ]}
84  @ ˜41& B2Nos vemos en la riviera te
   llamo de Monaco, zarpando de Malta ...

c:\Users\stconh\Documents\
   FinalProjectCrypto\CPEN503>
```

III. Conclusion

In this article we explored, amongst other things, a very important maxim of computer security: That any and all systems or schemata is vulnerable. Particularly we exposed, analytically and computationally, the flaws pertaining to Pfleeger & Pfleeger's dual asymmetric schema, and its man-in-the-middle attack thereof. Albeit notwithstanding reverence to proper mathematical acta and rigorous computational testing, nonetheless was shown, without reasonable doubt, that Pfleeger & Pfleeger's as pertaining to securement of key distribution is *least sufficiently flawed*; the probability than an attacker ought successfully exercise the attack is a subject of further research. Comprehensibly the equity of an attack could be defined as the difference of the value of the assets of information and the liability of the cryptanalyst system. In the particular scenario discussed, we attained finding of a way to describe a vulnerability *of* the system; however this might not always be possible. In conclusion, the only present way to minimize vulnerabilities of persons in the middle or masquerading the third party is to eliminate the trusted third party (at the cost of increased expenditure in processing power). We know only of one such algorithmic schemata: That which conjoins a blockchain with a peer-to-peer network [4].

Acknowledgment

This work is dedicated in memory of Dr. Javier Monserrate-Mills. It's also dedicated to my parents, siblings, and family, for infinite reasons throughout. More so, the author's dearly humbled and dedicates establishment of creativity in precedence and for future remembrance of the P. family, II & III established. In witness whereof in this city of San Juan, Puerto Rico, 1215 MMXVI Gregorian.

References

[1] C. P. Pfleeger and S. L. Pfleeger, *Analyzing Computer Security: A Threat / Vulnerability / Countermeasure Approach*. Prentice Hall, 2011.

[2] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal, Vol 28, pp. 656715*, Oktober 1949.

[3] W. Stallings, *Cryptography and Network Security: Principles and Practice (6th Edition)*. Pearson, 2013.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf."

Manuel Carro is a graduate student in the department of electrical and computer engineering at Universidad del Turabo, SUAGM. He is Principal at Atelier Limited, an Antillean technology company. Carro is a graduate of the University of Dayton's School of Engineering and the Institute for Shipboard Education, and coursed studies at the University of Puerto Rico Mayagüez campus. A native of Orocovis, he's also a trustee of Student Globe Corporation.