ABSTRACT

Title of thesis: CHAUM'S PROTOCOL FOR DETECTING MAN-IN-THE-MIDDLE: EXPLANATION AND DISCUSSION

William Newton, Master of Computer Science, 2010

Thesis directed by: Dr. Alan T. Sherman
Department of Computer Science

In this research paper, I explain David Chaum's patent that describes the Man-In-The-Middle (MITM) detection protocol. The MITM Detection Protocol (MDP) uses three stages to entrap an adversary that reveals her existence based on contextual information from each scenario. One of the primary goals of this research is to develop a detailed understanding of Chaum's novel concept since the language describing the MDP presented a clouded explanation. My second goal is to re-describe the protocol using conventional notation and illustrations for two cases in each scenario, with and without an adversary. The explicit illustrations provided within this paper clearly define the exchange between two communicants and their ability to detect an adversary. I define the assumptions based around the common random string model that eliminates prior exchange of information and distribution of shared secrets. A discussion follows the explanation of the protocol which concludes that Chaum's MDP adds technical value and merit towards achieving a MITM detection mechanism based on a restrictive assumption set; however, the

elevated complexity experienced by the user contributed to the lack of integration and acceptance of the MDP onto existing hardware solutions.

CHAUM'S PROTOCOL FOR DETECTING MAN-IN-THE-MIDDLE: EXPLANATION AND DISCUSSION

by

William Newton

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County in partial fulfillment
of the requirements for the degree of
Master of Computer Science
2010

Advisory Committee:
Dr. Sherman T. Alan, Chair/Advisor
Dr. John Pinkston
Dr. Dhananjay S. Phatak
Dr. Chintan Patel

Acknowledgments

First, I would like to thank my adviser, Dr. Alan T. Sherman for giving me an opportunity to work on a challenge that was difficult, but yet exciting, over the past two years. His experience and guidance has been invaluable toward my success as a researcher at UMBC. Second, I would like to thank David Chaum for the opportunity to review his patent application on the man-in-the-middle detection protocol. Furthermore, I am grateful for his insight and guidance as I gained an understanding of the concepts outlined in the patent application. Next, I would like to thank all the members of the Cyber Defense Lab at UMBC for their valuable feedback during my presentations on the protocol description. Lastly, I would like to thank my wife who has been very patient as I conducted my research, even thought it took up a lot of my time previously spent with her and my two daughters.

Table of Contents

List of Figures v

List of Abbreviations vi

1 Introduction 1
- 1.1 Introduction to the MITM 1

2 Overview 3
- 2.1 Protocol Overview 3
- 2.2 Scenario 1: Application for Real-Time Voice or Video 4
- 2.3 Scenario 2: Application for Real-Time Voice or Video 5
- 2.4 Scenario 3: Application for Text Messaging, Videomail or Voicemail . 5
- 2.5 Scenario 4: Application for Real-Time Voice or Video 6
- 2.6 Scenario 5: Mutual Discovery and Authentication Through a Friend . 7
- 2.7 Scenario 6: Privacy Enhanced Mutual Discovery and Authentication Through a Friend 7
- 2.8 Scenario 7: Friend of a Friend Discovery and Authentication 8

3 Model 9
- 3.1 Overview 9
- 3.2 Assumptions 9

4 Previous Work 12
- 4.1 Public Key Fingerprint 12
- 4.2 Interlock Protocol 13
- 4.3 Zimmerman Zfone 14

5 Stage Two : The Commitment Stage 16
- 5.1 Explaination 16
- 5.2 Introduction of Eve 18

6 Stage Three : The Detection Stage 21
- 6.1 Overview 21
- 6.2 Scenario 1 21
- 6.3 Scenario 2 24
- 6.4 Scenario 3 27
- 6.5 Scenario 4 30

7 Friend-of-Friend Scenarios 33
- 7.1 Overview 33
- 7.2 Proof Subroutine 34
- 7.3 Scenario 5 38
- 7.4 Scenario 6 39
- 7.5 Scenario 7 40

8 Discussion 42

9 Conclusion 46

Bibliography 48

List of Figures

5.1 Stage 2, The Commitment Stage. 17
5.2 Stage 2: Introduction of Eve. 19

6.1 Scenario 1 : Live Audio/Video Streaming 22
6.2 Scenario 1 : Live Audio/Video Streaming with Eve Present 24
6.3 Scenario 2 : Live Audio / Video Streaming 25
6.4 Scenario 2 : Live Audio / Video Streaming with Eve 26
6.5 Scenario 3 : Video-mail / Voice-mail 28
6.6 Scenario 3 : Test Messaging, Videomail, Voicemail with Eve 29
6.7 Scenario 4 : Live Voice / Video Streaming 31
6.8 Scenario 4 : Live Voice / Video Streaming with Eve 32

7.1 Proof Subroutine 35
7.2 Proof Subroutine with Eve 37
7.3 Scenario 5 39
7.4 Scenario 6 40
7.5 Scenario 7 41

List of Abbreviations

CRS	Common Random String
MDP	MITM Detection Protocol
MITM	Man-in-the-Middle
SSH	Secure Shell
SSL	Secure Sockets Layer

Chapter 1

Introduction

1.1 Introduction to the MITM

The Man-in-the-Middle (MITM) attack is a form of active eavesdropping by an adversary who acts as a broker, relaying information from one victim to the next. The attack violates all three information assurance principles: confidentiality, integrity and availability. With direct access to the information in transit, the adversary can sniff, analyze and modify messages without detection [1]. David Chaum submitted a patent application describing a MITM Detection Protocol (MDP); however, his explanation presented a blurred view on the details of the protocol. In this paper, I describe the MDP with the aid of illustrations to clearly define the exchange between two communicants with and without an adversary present.

Chaum's MDP is a scenario-based protocol that enhances the characteristics of the communication environment to detect an adversary. The MDP provides several new and significant contributions while building on existing research, utilizing the Common Random String (CRS) model to detect an adversary. The CRS model implies a less restrictive assumption set without the need to exchange information prior to communication. The MDP protocol was designed to use existing hardware and software components; as a result, the MDP could be deployed onto hardware platforms with a simple software update. Finally, the MDP asserts the assumption that an adversary cannot *think* like her victims, but rather, she can *act* like her victims. While this assumption is subtle, there is a distinct difference between the MDP and other detection solutions that impose additional assumptions on the adversary's behavior [4].

The MDP was written in a patent application using a dialog that exerted uncertainty and confusion with a concept that proved to be unique and challenging. In fact, five renowned cryptographic researchers accepted an all expense paid invitation to meet with David Chaum where he described the MDP face-to-face. By the end of the day, four of the five experts still did not grasp the concept!

Existing research and software solutions provide several techniques to detect a MITM adversary. The public key fingerprint solution, most notably used by the SSH client, uses an out-of-band secure channel to pre-exchange information. Rivest's interlock protocol and Zimmerman's Zfone application use voice recognition to identify an active MITM attack. Unlike previous approaches to detect an adversary, the MDP does not rely on shared secrets, out-of-band communication exchanges or the use of voice recognition. The research presented in this paper will provide a complete description of the MDP for each scenario defined in the patent application.

Chapter 2

Overview

2.1 Protocol Overview

The MITM Detection Protocol is split into three stages. During the first stage, known as the Session Key Exchange stage, Alice and Bob[1] derive a session key using the Diffie-Hellman (D-H) key exchange algorithm.[2] Without a trusted public key exchange mechanism, the D-H key exchange is vulnerable to a MITM attack. Consequently, Alice and Bob are not certain if they are victims of a MITM attack. When Eve is present, Alice and Bob will misinterpret each others public key with Eve's public key resulting in compromised information. Despite Eve's presence, Alice and Bob move to stage two, the Commitment stage, in an effort to detect Eve. From this point forward, information sent from Alice, Bob or Eve shall be encrypted with their derived session key.

Stage two is identified as the Commitment stage where Alice forces Eve to modify her specially crafted message that contains a committed value coupled with their public cryptographic keys. If Eve decides to forward Alice's original message, then Eve's presence would be detected with certainty. If Eve desires to avoid detection, she must recreate Alice's special message coupling with Eve and Bob's public keys. However, without knowing Alice's value before commitment, Eve is forced to *guess* and commit a value. Alice eventually reveals her original value to Eve. To maintain consistency, Eve must reveal her *guessed* value to Bob. Using both original values, Alice and Bob compute y. If Alice and Bob hold identical values of y, then Eve ceases to exist; otherwise, Alice and Bob are communicating through Eve. In either case, neither Alice nor Bob know if their y values match at the end of stage two.

During the third stage, the Detection stage, Alice and Bob will use their y values acquired in stage two to *enhance* their communication environment. If Alice and Bob have identical values, then their environmental *enhancements* will remain consistent. However, if their values do not match, then Eve must translate messages from one communication environment to another to avoid detection. Due to physical limitations of each environment, Eve will not be able to sustain her message translation capabilities between her victims, thus being detected.

There are seven scenarios that can be applied in stage three. Each scenario provides a limitation where Eve's existence is detected.

2.2 Scenario 1: Application for Real-Time Voice or Video

Alice and Bob split their y value into four parts, y_1, y_2, y_3, and y_4. y_2 specifies an interval of time when *both* parties, Alice and Bob, can speak. y_4 specifies an interval of time when *neither* Alice nor Bob can speak. y_1 and y_3 specifies the interval of time when Bob and Alice can speak independently. With Eve present, the subvalues will be different from Alice and Bob's perspective. As a result Eve will not be able to maintain a consistent flow of communication while meeting the modified communication environment.

2.3 Scenario 2: Application for Real-Time Voice or Video

Alice and Bob split their y value into two parts, a salt value, y_1, and a database index, y_2. The database index is used to reference a specific element in a public database. For simplicity, a database of jokes will be used in this example. When Alice and Bob have identical database index values (y_2,) then they will retrieve a matching joke from the database. Next, Alice will create a verbal recording of her joke. The recording is encrypted with a cryptographic key (known only to Alice,) salted with her derived salt value, y_1. Alice sends the encrypted recording as the first message to Bob. Without having access to Alice's key, Bob must wait to decrypt the recording. Following the protocol, Bob will record the punchline to the joke as a response. After Alice receives Bob's response, she releases the unsalted version of the cryptographic key she used to encrypt her joke. Bob decrypts Alice's first message to reveal her joke. If Alice and Bob perceive a match between the joke and the punchline, then a MITM adversary does not exist.

2.4 Scenario 3: Application for Text Messaging, Videomail or Voicemail

Alice and Bob communicate in a ping-pong-like style, each producing a response from a previous request. Both parties split their y value into two parts, a salt value, y_1, and a time interval, y_2. Alice creates and encrypts her message with a random cryptographic key salted with the derived salt value. After Bob receives the encrypted message, he pauses for the duration specified by the time interval. Afterwords, he responds with a cryptographic key request message prompting Alice to release the unsalted cryptographic key. Bob applies his derived salt value with Alice's key to decrypt her message. The process is repeated where the roles are reversed. Eve is detected upon matching one of two conditions: The key request message is received outside a grace time interval. The lifespan of encrypted messages fail to exceed half of the total time elapse.

2.5 Scenario 4: Application for Real-Time Voice or Video

Real-Time applications rely on a constant stream of packets that meet timing characteristics to maintain a quality of service. Alice monitors the time intervals on incoming packets. Once in a while, Alice tags an outgoing packet using a publicly known technique. Bob receives and encrypts the marked packet with a random private cryptographic key salted with y. The encrypted packet is returned to Alice. Bob reveals the unsalted key in the proceeding message. Alice uses a statistical toolkit to determine the expected delivery time of the marked packet. If the packet's round trip time exceeds the expected time interval, then Eve existence is known. The incurred delay is introduced by Eve when she re-encrypts the marked packet from one session key to another; however, Eve must wait until Bob reveals the random cryptographic key, thus adding a delay.

2.6 Scenario 5: Mutual Discovery and Authentication Through a Friend

Alice and Bob exchange authentication credentials based on an existing friendship with a mutual friend. Authenticators provide proof of an existing, genuine friendship. Alice and Bob exchange a set of authenticators to determine if they have a friend in common. Once a common friend is established, Alice and Bob prove to each other that the authenticators they forwarded are genuine following a simple *proof*[3] routine. If Alice or Bob detect that the authenticators are not genuine, then Eve was detected.

2.7 Scenario 6: Privacy Enhanced Mutual Discovery and Authentication Through a Friend

Alice and Bob exchange authentication credentials based on existing friendships with a mutual friend without revealing the identity of their mutual friend. Alice and Bob use random exponents to obfuscate their friend's identify in their authenticators. Using basic algebra techniques, Alice and Bob can derive a common set of matching authenticators where they are assured they have a friend in common. Similarly in the previous scenario, Alice and Bob follow the 'proof' subroutine to prove the authenticators are genuine. Eve's detection is centered around illegitimate authenticators.

2.8 Scenario 7: Friend of a Friend Discovery and Authentication

Alice and Bob exchange authentication credentials based on a friend-of-a-friend relationship. Similarly as in the past two scenarios, authenticators are used to validate friendships between Alice and Bob. If Eve exists, her presence will be detected in the 'proof' subroutine.

Chapter 3

Model

3.1 Overview

The MITM Detection Protocol follows the common random string model. Predistributed information and shared secrets are not assumed in the MDP. Without a secure authentication mechanism, a MITM adversary can simultaneously impersonate two parties while violating confidentiality and integrity information assurance concepts. As a result, the creation of the session key in the first stage of the MDP is vulnerable towards a MITM attack.

The MDP can be deployed on a variety of computational devices from laptops to smart phones. Each communicant may independently choose their MDP enabled device that fulfills the scenario requirements. For example, if the scenario depends on a voice transmission, then the MDP enabled device should contain voice recording support. The MITM adversary is not limited to a specific platform. The MDP will defend against any adversary using unrestricted resources.

3.2 Assumptions

The MDP asserts several assumptions that outline the behavior and characteristics between Alice, Bob, and Eve defined as:

- *Key Assumption:* The MITM adversary cannot *think* like her victims. For example, the adversary cannot generate original content on behalf of her victims.

- *Key Assumption:* Public keys are not authenticated, pre-distributed or referenced from a trusted resource.

- *Key Assumption:* Shared secrets are not used by the communicating parties. Information shared between two communicating parties will be publicly known by all parties.

- The adversary can impersonate her victims using a separate set of public/private keys.

- Secure computational devices shall be used by each party.

- Each computational device contains a cryptographic one-way function with the following properties:

  - Given a one-way function and a message, it is easy to compute an image.

  - Given an image, it is computationally difficult to determine the original message.

  - It is computationally difficult to modify the original message without modifying the image.

  - It is computationally difficult to find two messages that compute to the same image.

The three designated *key* assumptions differentiate the MDP from other MITM detection or prevention protocols. The first and most subtle key assumption asserts the ability that Eve can *act* like her victims, but she cannot *think* like her victims. Therefore, Eve can replay messages from one party to another, but she cannot generate original content on ones behalf without being detected.

Chapter 4

Previous Work

Man-in-the-middle attacks, also known as bucket brigade attacks, have been documented in existing protocols such as the Needham-Schroeder protocol [2]. There are a variety of MITM detection or prevention mechanisms to catch an adversary before sensitive information is exchanged. For example, Gavin Lowe described in [2] a MITM prevention mechanism that is unique to the Needham-Schroder protocol. However, a strict assumption set imposed by each detection mechanism limit their applicable usefulness in a practical environment. The MDP uses a less stringent assumption set than those found in other solutions, including public key fingerprints, the interlock protocol, and the Zfone.

4.1 Public Key Fingerprint

Public key fingerprints provide a defense against MITM attacks using public key cryptology. Each communicating party creates a hash image of their public key called a fingerprint. Using an out-of-band secure channel, both parties exchange their fingerprints for future reference. In subsequent communication exchanges, each party can validate their opponent's public key using the cached fingerprint.[5].

The fingerprint solution relies on the assumption that information must be exchanged over a secure channel prior to communication. The MDP uses a less restrictive model where prior information exchange, authentication of public keys, and shared secrets are not needed to detect a MITM adversary. As a result, the MDP may be utilized in broad array of scenarios.

4.2 Interlock Protocol

Rivest and Shamir developed another approach to detect MITM adversaries with the interlock protocol. The interlock protocol was designed to authenticate two parties, given their ability to recognize each other's voices. Furthermore, both parties do not depend on a secure centralized public key infrastructure, nor do they need to exchange information prior to communication [6].

The interlock protocol is split into two stages, the key exchange stage and the detection stage. The former stage uses the Diffie-Hellman key exchange protocol to establish a session key between two parties. During the latter stage, both parties split their first encrypted voice message into two parts. The first message part is exchanged between both parties followed by the exchange of the second message part. The message maintains its encrypted state until all the parts are received. Therefore, the adversary must receive both parts of the encrypted message to re-encrypt the message from one session key to another. Since the message arrives in parts, the adversary must forward or *fake* a message to one of her victims, ultimately leading to her detection.

An attack was constructed to exploit the timing characteristics of the interlock protocol[3]. A solution that safeguarded against the timing attack used a technique called forced-latency modification[7]. Both variations on the protocol - the interlock protocol and the forced-latency interlock protocol - do not rely on predistributed information or an authenticated key exchange server to validate public keys, similarly to the MDP. However, the interlock protocol assumes that both parties have the ability to recognize each other's voices. The MDP does not rely on voice recognition to detect an adversary for every scenario. Furthermore, the MDP assumes that an adversary cannot *think* like her victims without revealing herself; however, an adversary can *act like* or impersonate her victims. The subtle difference in the assumption set differentiates the MDP from the interlock protocol.

4.3 Zimmerman Zfone

Zfone is another project aimed to secure VoIP communications led by Phil Zimmermann. Zfone is designed to use ZRTP[4], a secure protocol for real-time applications. ZRTP uses three levels of protection to provide secure communications. The first level uses a hash image of their public keys as an index into a publicly known dictionary of words. Both parties verbally communicate their word while recognizing their communicant's voice to validate that a MITM adversary does not exist. The second level of protection builds upon information stored from a previous engagement. The final level of protection relies on a public key infrastructure to provide integrity controls for the delivery of SIP messages [8].

The Zfone project does not rely on predistributed information or an authenticated key exchange server to validate public keys. However, the Zfone assumes that both parties must conduct voice recognition to validate the communicant's voice. As described in the last section, the MDP does not rely on voice recognition to detect an adversary for every scenario. The Zfone provides an extensive security posture based on the three levels of protection. The third level of protection would require additional hardware/firmware and software updates to safeguard the private components on a user's device. For example, locking private cryptographic keys on a cellular phone may require additional modification to the hardware for protected access. The MDP was designed to used existing hardware solutions with a minor software update.

Chapter 5

Stage Two : The Commitment Stage

5.1 Explaination

The MDP is split into three stages. The first stage, the session key exchange stage, establishes a session key to encrypt communication sent between two communicants. The adversary may exploit the communication while Alice and Bob create a session key based on the noted assumptions. All communications following the first stage remain encrypted using the derived session key. The commitment stage in the MDP entraps an adversary to make decisions that are detectable in the third and final stage.

The commitment stage forces Eve to make a blind decision detectable by Alice and/or Bob. As shown in Figure 5.1, Alice generates a random value, x. Then Alice applies a one-way function to create an image using her own public key, Bob's public key and the random value, noted as $f(x, q^c, q^d)$. Alice forwards the image to Bob without revealing her random value, x. Bob saves the image he receives presumably from Alice as V. Then Bob generates and forwards a random value x' as a response to Alice. Both random values, x and x' are independently generated.

Alice saves Bob's randomly generated value, x'. Finally, she reveals x in her final message in stage two. Bob has all the parameters (Alice's random value x, Alice's public key, and his own public key) to recreate Alice's image to validate

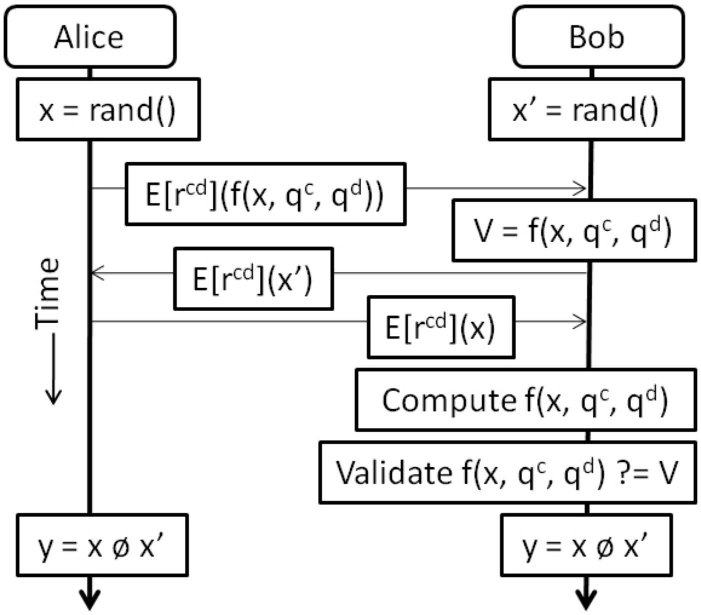


Figure 5.1: Stage 2, The Commitment Stage.

for equality against V. If the validation fails, Eve's presence has been detected. However, if the validation succeeds, Alice and Bob are not certain if an adversary exists. Assuming the image verification is correct, Alice and Bob compute y and y' respectively.

The computation of each y value must be influenced by the randomly generated values, x and x'. The computation should be known by all participants, including the adversary. Furthermore, the computation may be as complex or simplistic as desired. For example, y could be calculated by concatenating x and x'. As referenced in this document, the function will be represented by the ø symbol. (i.e. $y = x ø x'$.)

If Alice and Bob derive identical y values, then their communication would be considered secure from a MITM attack. However, if their values are inconsistent, then a MITM adversary would exist. In either case, Alice and Bob need to share their corresponding y values using techniques that would restrict any influence by Eve. The techniques to share each y value are defined in stage three.

5.2 Introduction of Eve

At the conclusion of stage one, Eve poses an imminent threat as the MITM adversary between Alice and Bob's conversation. Session keys established between Alice and Eve and between Eve and Bob. Eve attempts to avoid detection by re-encrypting messages from one session key to another to maintain a seamless flow of communication between Alice and Bob. Eve could choose to modify messages, however doing so would increase the possibility of detection. Eve's goal is to retain her stealthy status modifying messages to avoid detection.

The commitment stage forces Eve to make blind decisions that are irreversible and detectable. Alice sends the image, $f(x, q^c, q^f)$, as the first message of the commitment stage. When Eve receives the message, she is forced to make her first decision: Eve could forward Alice's image without modification or Eve could generate (or pick) a value x'' and recompute the image, $f(x'', q^f, q^d)$. In the former case, Bob will discover Eve existence after he validates his calculated image against V.

$$f(x, q^f, q^d) \neq f(x, q^c, q^f)$$

The latter case is shown in Figure 5.2.

Bob receives and saves Eve's new image as V for validation. Bob follows the

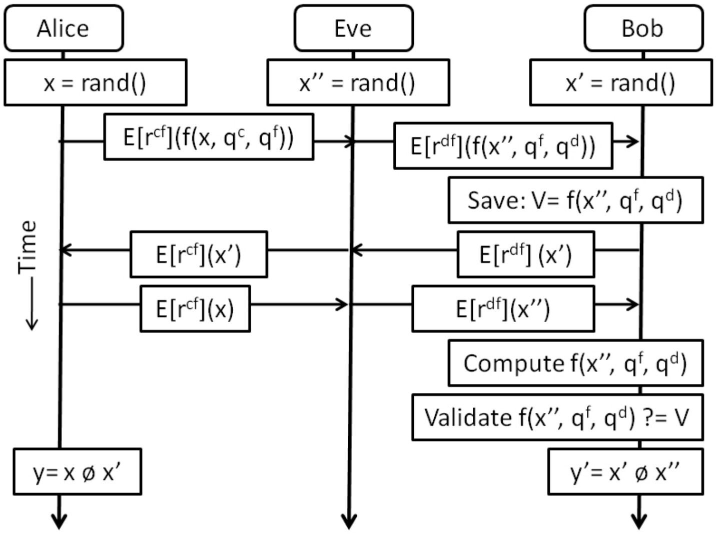

Alice
Eve
Bob
x = rand()
x'' = rand()
x' = rand()
$E[r^{cf}](f(x, q^c, q^f))$
$E[r^{df}](f(x'', q^f, q^d))$
Save: $V= f(x'', q^f, q^d)$
Time
$E[r^{cf}](x')$
$E[r^{df}] (x')$
$E[r^{cf}](x)$
$E[r^{df}](x'')$
Compute $f(x'', q^f, q^d)$
Validate $f(x'', q^f, q^d)$?= V
$y= x \o x'$
$y'= x' \o x''$

Figure 5.2: Stage 2: Introduction of Eve.

protocol and returns a random value x'. Eve forwards Bob's random value to Alice without modification. Then Alice releases her random value x in her final message. To maintain consistency, Eve must modify Alice's message by replacing x with x'' as the random value. Bob recomputes the an image using Eve's random value, Eve's public key and his public key for validation against V. The validation succeeds.

Finally, Alice, Bob, and Eve determine their respective y values. Alice computes y where $y = x\o x'$. Bob computes y' where $y' = x'\o x''$. Eve computes both y and y' since she is communicating to both Alice and Bob concurrently. Since Eve is acting as the MITM adversary, Alice's computed value, y, does *not* match Bob's computed value, y'. However, Alice and Bob are unaware that their y values do not match. Stage three, the detection stage, illustrates several methods to convey their derived y values without Eve's influence.

Stage one defines the mechanism to establish a session key between Alice and Bob. Stage two entraps the adversary by forcing her to make *bad* decisions. Both stages must maintain the following property: The parameters used to create a session key in stage one should be captured in the image created by Alice in her first message in stage two. For example, the D-H key exchange protocol was used in stage one which uses Alice and Bob's public keys to create a session key. Their public keys must be included in the image created by Alice ($f(x, q^c, q^d)$) in her first message in stage two. Otherwise, Eve could forward Alice's first message while avoiding detection.

Chapter 6

Stage Three : The Detection Stage

6.1 Overview

In stage three, the detection stage, unique characteristics of the communication environment are utilized to detect an MITM adversary in a variety of scenarios such as video and audio communication, text messaging, email, voicemail and videomail. The characteristics of each scenario are slightly modified to detect the adversary. When a MITM adversary exists, Alice and Bob will derive y values that do not match. As a result, they enhance their communication environment based on different criteria. Eve must maintain a fluid stream of communication between Alice and Bob using two separate communication environments. Her inability to maintain a seamless connection will reveal her existence.

6.2 Scenario 1

Scenario one uses techniques that can be applied in live audio or video communications. At the beginning of stage three, Alice and Bob split their y values into four sub-components, y_1, y_2, y_3, and y_4. Each sub-component represents a time interval measured in seconds. Next, Alice and Bob must complete a four step communication exchange before they can communicate freely. The duration of each step is designated by the sub-component time intervals derived from y. Figure 6.1 illistrates each of the four steps.

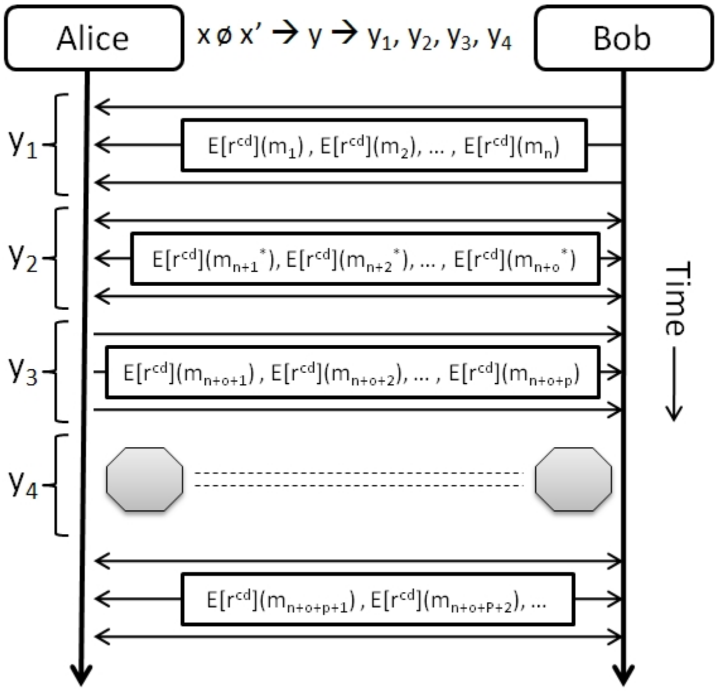

Alice
x ø x' → y → y_1, y_2, y_3, y_4
Bob
y_1
$E[r^{cd}](m_1)$, $E[r^{cd}](m_2)$, … , $E[r^{cd}](m_n)$
y_2
$E[r^{cd}](m_{n+1}^*)$, $E[r^{cd}](m_{n+2}^*)$, … , $E[r^{cd}](m_{n+o}^*)$
Time
y_3
$E[r^{cd}](m_{n+o+1})$, $E[r^{cd}](m_{n+o+2})$, … , $E[r^{cd}](m_{n+o+p})$
y_4
$E[r^{cd}](m_{n+o+p+1})$, $E[r^{cd}](m_{n+o+P+2})$, …

Figure 6.1: Scenario 1 : Live Audio/Video Streaming

The first step designates the communication flow from Bob to Alice for the time interval y_1. During this period of time, all visual/audio content from Alice should be suppressed. The second step designates a bi-direction communication flow between Alice and Bob for the time interval y_2. Step three specifies the communication flow from Alice to Bob for the time interval y_3. Following Alice's example during the first step, all visual/audio content from Bob should be suppressed. Finally, all communication should cease between Alice and Bob for the time interval y_4 in the fourth step. If the observed communication flow meets Alice and Bob's expectation based on each time interval, then a MITM adversary does not exist.

When Eve is embedded in the communication link between Alice and Bob, then their derived y values will not match. As a result, Alice's expectation of the time interval for each step will not match Bob's expectations. Eve's goal is to mask her existence. Therefore, she will attempt to simulate communication between Alice and Bob. Eve will have the ability to slice, delay and truncate messages in transit to avoid detection. However, since Eve can not *think* like Alice or Bob and generate new content without being detected, Eve must exercise caution since excessive modification of the message control may unveil her presence. Figure 6.2 illustrates Eve's interaction between Alice and Bob.

Bob initiates communication with Alice through Eve in the first step as shown in Figure 6.2. As Alice moves onto step two, Bob remains in step one. Eve delays Alice's communication until Bob enters step two. It is possible that Bob would notice the disjointed discussion presented by Alice when he receives her message several seconds later. (i.e. Alice commenting on a remark by Bob earlier in the conversation.) Since the conversation is plausible, it is assumed Bob does not suspect suspicious activity yet.

Alice and Bob continue to engage through each step while Eve continues to manipulate the stream between her victims. Finally, Bob detects Eve in step three. Eve delayed Alice's transmission in her instance of step three for Bob. However, Bob's time interval for step three is longer than Alice's transmission. Without the ability to generate new content, Eve becomes revealed. As a result, Bob terminates his connection with Alice.

Eve can not generate content on behalf of her victims, however, she can re-

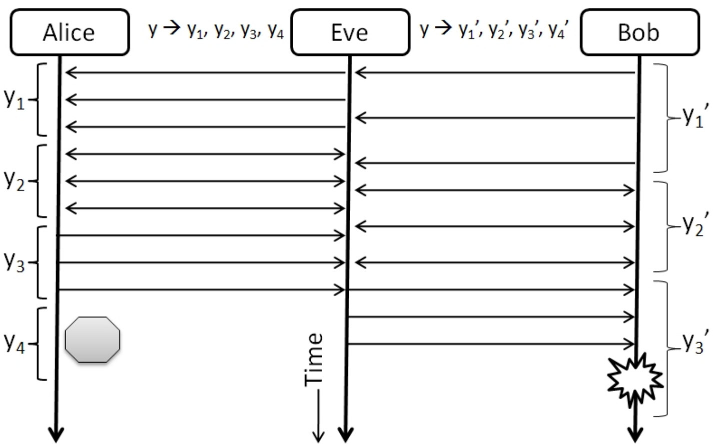


Figure 6.2: Scenario 1 : Live Audio/Video Streaming with Eve Present

play content to fulfill the desired time interval that meets Bob's expectations. The disjoint conversation would generate obvious suspicion that the communication link has been tampered, revealing Eve's existance.

6.3 Scenario 2

Scenario two exhibits another technique to detect a MITM adversary for live audio or video communication applications. Figure 6.3 illustrates the communication exchange between Alice and Bob in stage three.

Following the Commitment stage, Alice and Bob split their y value into two parts, a salt value, y_1, and a database index, y_2. The database index is used to reference a specific element in a public database, W. For simplicity, a database of jokes will be used in this example. Alice creates a random key, K, salted with y_1, symbolized as K^{y1}. Both parties, Alice and Bob, reference a joke in the public database, W, using the index y_2. Alice records her joke as the first message, m_1.

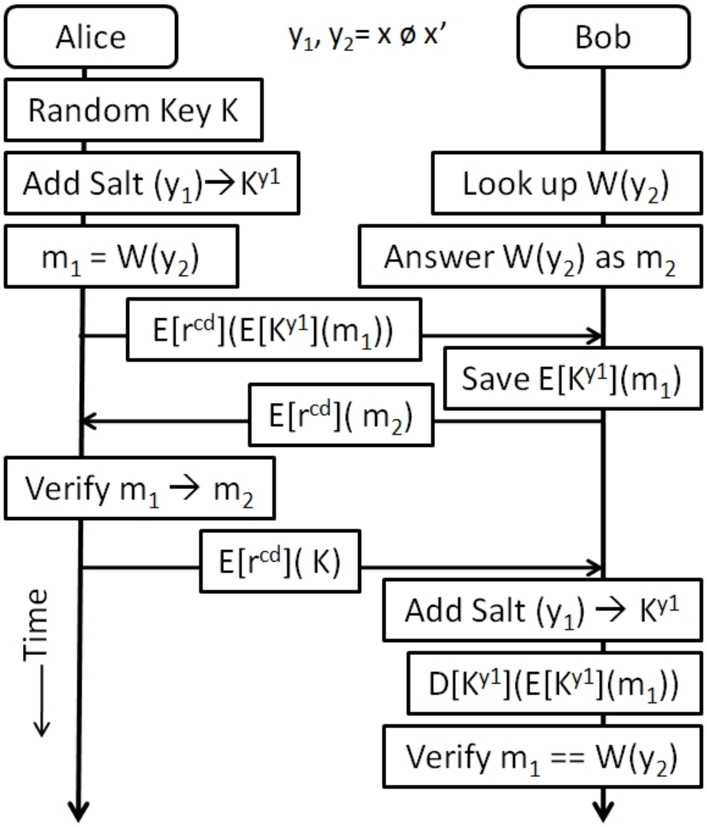


Figure 6.3: Scenario 2 : Live Audio / Video Streaming

Bob records the punch line to his joke as the second message, m_2. After encrypting m_1 with the salted key, Alice sends Bob her recorded message. Without knowledge of K, Bob is not able to listen to the recording, therefore, he saves the message for reference. Bob responds with the second message. Upon receipt of m_2, Alice release the key, K, in the third message. Finally, Bob can apply his derived salt value y_1 to K to decrypt Alice's first message. Three conditions must be satified before Alice and Bob are convinced that a MITM adversary does not exist:

- Alice verifies that the punch line she receives as the second message, m_2, matches her original joke.

- Bob decrypts Alice's first message successfully.
- Bob verifies that the joke he received from Alice matches his punch line.

Upon any failed condition, the connection is terminated due to the detection of Eve.

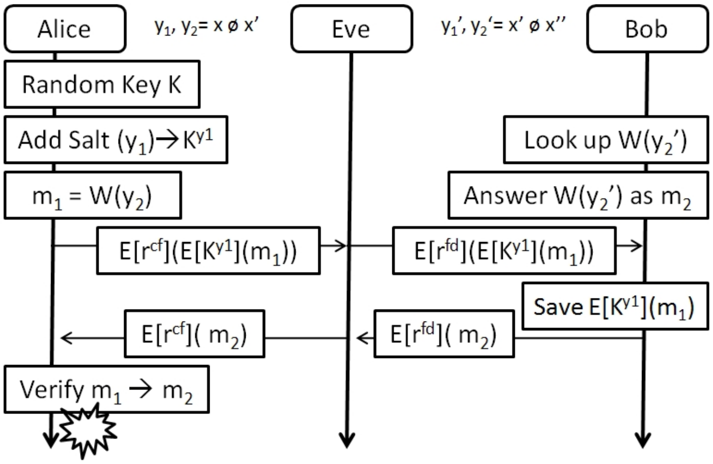


Figure 6.4: Scenario 2 : Live Audio / Video Streaming with Eve

Figure 6.4 illustrates the effect when Eve intercepts the communication channel between Alice and Bob. As a result, Alice and Bob compute different y values at the end of stage two. Alice references a joke in the database using index y_2. However, Bob will reference a different joke using index y_2'.

Alice creates a voice recording of her joke, encrypts the message with K^{y1}, and sends the message to Eve. Eve has a decision to forward Alice's message to Bob, retransmit a previously captured joke created by Alice or immediately respond to Alice with a previously captured "punch line" provided by Bob. In either case, Alice will detect Eve before releasing her key, K. Using the former decision, Bob saves Alice's encrypted joke and responds with his punch line in the next message. Eve continues to forward all messages from one party to the other without modification.

Before Alice releases K, she must verify that the punch line matches her joke. With different database indexes, the verificaton check fails. As a result, Alice terminates the connection with Eve without releasing the key, K. Without the ability to continue a genuine conversation, Eve is forced to terminate her connection with Bob.

6.4 Scenario 3

Scenario three provides a MITM detection technique for text messaging, voicemail and videomail applications. In each application, Alice and Bob communicate in a ping-pong-like fashion where they must take turns communicating. Alice initiates the exchange, as illustrated in Figure 6.5, for the duration of the first round. Then the process is repeated in round two where Bob authors the second message.

Before both parties exchange any messages, Alice and Bob split their y values into two subcomponents, a salt value, y_1, and a time interval, y_2. Alice generates a message labeled as m_1. Then, Alice generates a random key, K salted with y_1, symbolized as K_{y1}. Alice encrypts her message with the salted key before transmitting it to Bob. Bob, or a system acting on Bob's behalf, waits the designated time specified by y_2 before responding with a key request message. Alice waits for Bob's request ensuring that adequate time elapses before releasing the unsalted key, K. Finally, Bob salts the key to decrypt Alice's original message. If the decryption fails, then a MITM attack is evident.

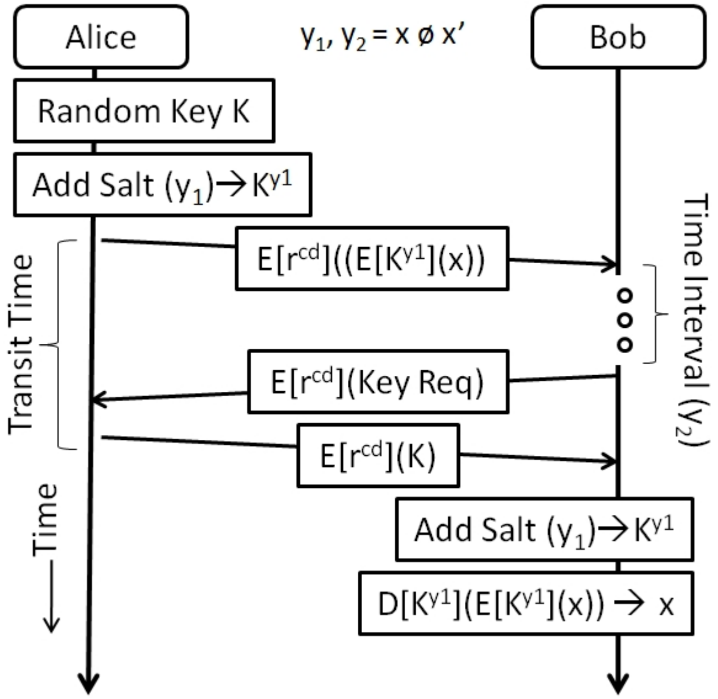


Figure 6.5: Scenario 3 : Video-mail / Voice-mail

Another condition must be satisfied before preceding to the second round. The ”transit time” must be much longer than half the ”total round time.” Transit time is measured by the sender as the time elapsed between the transmission of the first message and the key, K. Total round time is measured as the time elapsed in one round. If the transit time fails to exceed half the time elapsed for a specific round, then Eve was detected.

Figure 6.6 illistrates Eve's engagement in the first round of communication between Alice and Bob. Alice sends her first message encrypted with K^{y1}. Eve receives Alice's message and waits the designated time, y_2 before replying with a key request message. Alice releases the unsalted key, K, in the next message. Eve reencrypts Alice's message using K with Bob's salt value y_1' and forwards the result to Bob. Bob waits for the time interval, y_2' before sending a key request message. Eve releases K when she receives Bob's key request. Finally, Bob can decrypt Alice's original message.

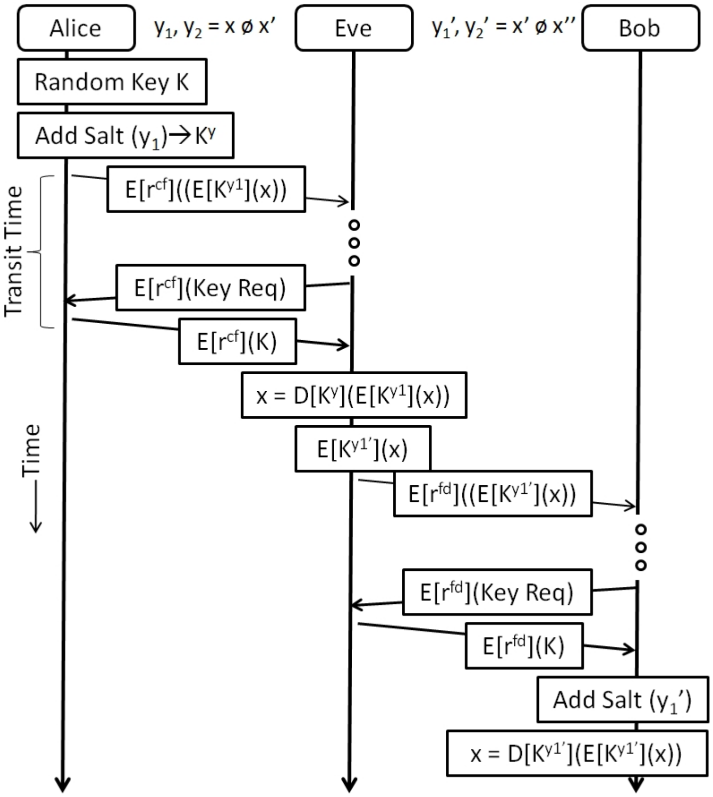


Figure 6.6: Scenario 3 : Test Messaging, Videomail, Voicemail with Eve

After Alice sent her encryption key, she continued to monitor the total round time to compare against the recorded transit time. Eve was forced to engage in two message exchanges for each round of communication. As a result, the transit time recorded by Alice will be roughly half the total round time, a violation of the second condition which is an indication that Eve is present.

6.5 Scenario 4

The fourth scenario describes a MITM detection technique for real-time applications such as voice or video chat. This technique utilizes the inherent property that a constant stream of packets are transmitted that are constrained by time to meet jitter and latency requirements. If the latency or jitter values are too excessive, then Alice and Bob will notice a degradation in the communication signal. Figure 6.7 illustrates the real-time communication exchange between Alice and Bob. For simplicity, the Figure illustrates packets streaming from Alice to Bob followed by packets that stream from Bob to Alice. Realistically, packets are constantly flowing in both directions during the entire communication.

As illustrated in Figure 6.7, Alice is continuously transmitting packets to Bob. Occasionally, Alice will 'mark' a packet for Bob. Packets are marked using the following procedure:

- Alice generates a random key and applies a salt based on her derived y value. (K^y)
- Alice inserts a unique identifier in the content stream.
- Alice encrypts the packet containing the unique content using K^y and forwards the marked packet to Bob.

When Bob receives the encrypted packet, he saves it temporarily while he awaits for the key. Alice concatenated her key, K, in its unsalted form in the following packet. Finally, Bob may decrypt the content using K salted with y. Bob listens and then repeats Alice's message verbatim as part of the conversation. He encrypts the message using the salted key and forwards the message to Alice. When the encrypted message returns to Alice, Alice calculates the Key Message Interval (KMI.) The key message interval is the amount of time that elapsed between when the original message that was sent to Bob until the moment when Alice receives Bob's response. If the KMI matches the expected time interval, then the MITM adversary does not exist.

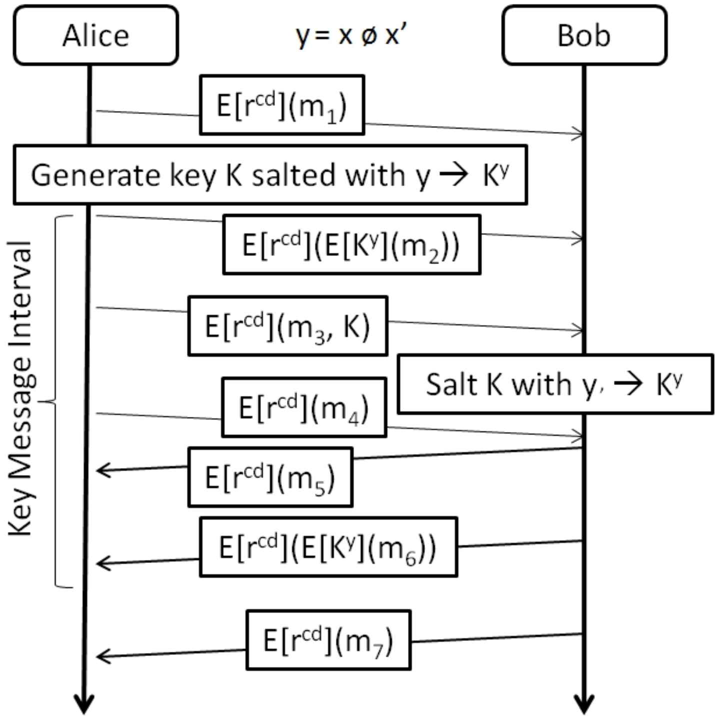


Figure 6.7: Scenario 4 : Live Voice / Video Streaming

When Eve exists as the MITM adversary, she will introduce an added delay to acquire the key from Alice before Eve can re-encrypted the message using the correct key/salt combination expected by Bob. As shown in Figure 6.8, Bob receives the encrypted message one unit time interval later than he did in Figure 6.7. However, Bob cannot detect the delay in transmission; therefore, Bob listens and repeats the message verbatim and sends the message back to Alice. Since the key is known by Eve on the reply, Eve decrypts the message from Bob to re-encrypt the message to Alice without adding a significant time delay. Alice detects the delay in transmission that was introduced by Eve when the message was enroute to Bob, thus uncovering Eve's presence.

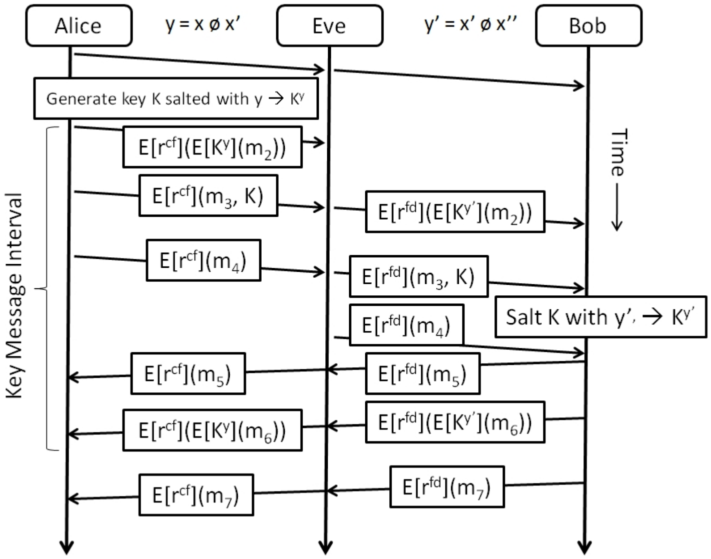


Figure 6.8: Scenario 4 : Live Voice / Video Streaming with Eve

Chapter 7

Friend-of-Friend Scenarios

7.1 Overview

Scenarios 5, 6, and 7, noted as 'friend-of-friend scenarios,' rely on a trusted relationship with a common friend between two parties who wish to communicate. Both parties may or may not be 'friends' themselves in the conventional definition of the word. Using a common friend, both parties can determine if a MITM adversary exists by proving their identity.

The friend-of-friend scenarios follow a different protocol structure than what was described as the three stages of the MDP. The first stage, the key exchange stage, is required to establish a session key between two parties. However, the second and third stages are merged together using a unique 'proof' subroutine to detect a MITM adversary. The subroutine uses the same commitment technique as described in stage two of the conventional protocol where the adversary is forced to make decisions that will reveal her existence.

Friend-of-friend scenarios use authenticators as proof of an existing relationship with a friend. Authenticators are assumed to be created over a secure communication link where a MITM adversary was not present. Authenticators use the w^{cd} notation where w signifies an authenticator between two or more people, represented by their private key exponent values, c and d. For example, if $c = 3$ and $d = 5$, then an authenticator between those two associated parties would be represented as $w^{(3*5)} = w^{15}$.For clarity, authenticators will explicitly show each private exponent variable to illustrate the relationship between two or more parties.

7.2 Proof Subroutine

All of the friend-of-friend scenarios use a 'proof' subroutine. The subroutine proves that the identity of one party has not been masked by an adversary using existing authenticators. The commitment technique illustrated in stage two of the conventional MDP is used in the subroutine to force the adversary to make a blind decision that exposes her existence. The subroutine provides a asymmetrical proof of one's identity. Therefore, one party fulfills the role of the 'prover' while the second party becomes the 'verifier.' The prover, represented by Alice in Figure 7.1 wants to prove to Bob, the verifier, her identity using the authenticators provided by Bob.

The proof subroutine is represented by the function, $proof(q, q^c, w, w^c)$, where:

- q is the base value used to construct public/private keys.
- q^c represents the public key of the prover.
- w represents the base authenticator.
- w^c represents the base authenticator raised to the private exponent held by the prover, called the challenge authenticator.

The proof subroutine begins where Alice computes a random value based on an iteration index i. Bob generates a random value between 0 and 1 based on

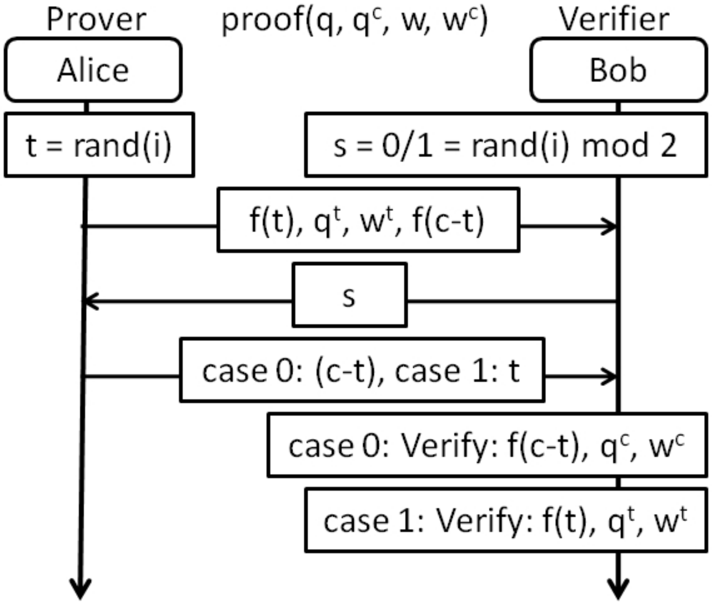


Figure 7.1: Proof Subroutine

the same iteration index i. Next, Alice computes the four data elements that will compose her first message to Bob. The first $(f(t))$ and fourth $(f(c-t))$ data elements uses a one-way function (f) as a commitment technique used to expose the adversary. The fourth data element is the image on the calculated difference between Alice's private exponent and her random value. The second (q^t) and third (w^t) data elements provide a verification set needed by Bob, the prover. Bob responds with the random value (0 or 1) that he generated. Alice's final message is determined by Bob's random value she received in the second message. Two cases exist:

- case 0: $(c-t)$
- case 1: t

After Bob receives Alice's response, he must validate three data values using the element provided by Alice. Bob must handle two cases. In the case where 0 was

Bob's random value, Bob must verify three elements. First, Bob must compute the image of $(c - t)$ to compare against the image he received in the first message from Alice. Second, Bob must compute Alice's public key in the following manner:

$$q^{(c-t)} * q^t = (q^c/q^t) * q^t = q^c$$

Bob verifies the calculated q^c matches the value given as the second parameter in the subroutine. Third, Bob must compute the challenged authenticator in the following manner:

$$w^{(c-t)} * w^t = (w^c/w^t) * w^t = w^c$$

Finally, Bob verifies the calculated w^c matches the value given as the fourth parameter in the subroutine.

In the case where 1 was Bob's random value, Bob must verify a different set of elements. First Bob must compute the image of t to compare against the image he received in the first message from Alice. Then, Bob must compute q^t and w^t to match against the values he received in the first message from Alice. In the former case, case 0, it is certain that an adversary does not exist. In the latter case, case 1, there is a possibility that the adversary was not detected. Therefore, the subroutine should be repeated while updating the iteration value, i, until both parties are confident that an adversary does not exist.

Introduction of Eve

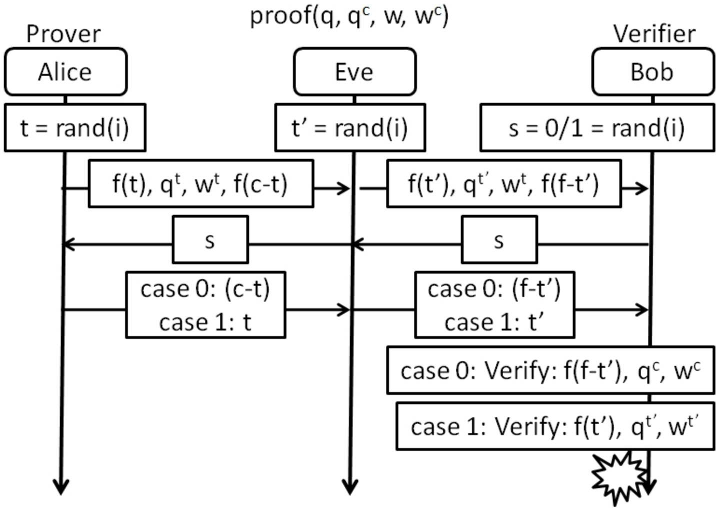

proof(q, q^c, w, w^c)
Prover
Alice
Eve
Verifier
Bob
t = rand(i)
t’ = rand(i)
s = 0/1 = rand(i)
f(t), q^t, w^t, f(c-t)
f(t’), $q^{t'}$, w^t, f(f-t’)
s
s
case 0: (c-t)
case 1: t
case 0: (f-t’)
case 1: t’
case 0: Verify: f(f-t’), q^c, w^c
case 1: Verify: f(t’), $q^{t'}$, $w^{t'}$

Figure 7.2: Proof Subroutine with Eve

Eve’s engagement in the proof subroutine shall be revealed by the verifier when the authenticator validation check fails. In Figure 7.2, The subroutine starts by generating a random value. Eve is forced to use a different random value than Alice due to the irreversible commitment image in Alice’s first message. Eve wants to forward Alice’s message; however, she must substitute her own random value in each data element to construct a message containing $f(t')$, $q^{t'}$, $w^{t'}$, and $f(f-t')$. Bob saves each data element he receives in the first message and responds with a random value, 0 or 1. Eve forwards Bob’s random value to Alice who responds with her final message containing $(c-t)$ for case 0 or t for case 1. Eve substitutes her random challenge authenticator in the following manner:

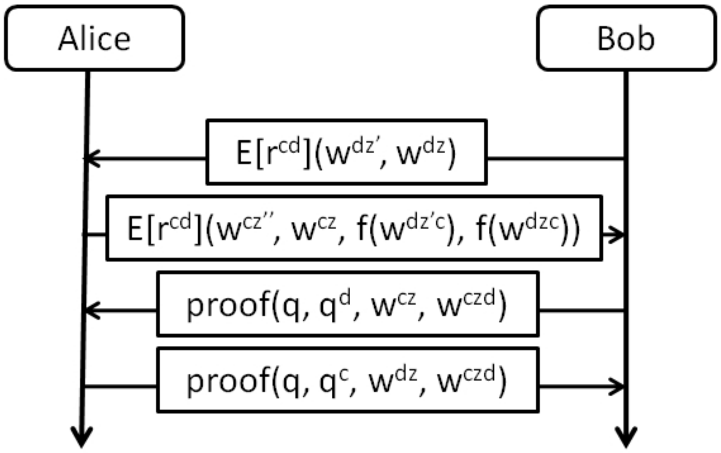
$$w^{(f-t')} * w^{t'} = (w^f/w^{t'}) * w^{t'} = w^f! = w^c$$

The final validation check fails, revealing Eve's presence. Eve cannot create, fake, or reproduce authenticators that prove a relationship between two foreign entities. As a result, Eve cannot modify the input parameter w^c to match Bob's expectation and avoid detection. The authenticator is tied to Alice's private exponent, restricting modification by Eve.

7.3 Scenario 5

Scenario five outlines an authentication protocol between two parties who have a friend in common. Figure 7.3 illustrates the communication exchange between Alice and Bob as they authenticate to each other. The scenario begins with the assumption that Alice and Bob have authenticators with a common friend. Bob initiates the communication exchange by sending a set of authenticators with friends who he believes they have in common. Alice responds with her set of authenticators in addition to an image of Bob's authenticators after they are *marked* using Alice's private exponent.

In the third 'message,' the proof subroutine is called by Bob as the prover. Bob proves his identity to Alice by marking one of Alice's authenticators (w^{cz}) who share a common friend outlined in the proof subroutine, thus obtaining w^{czd}. Upon verification of Bob's identity, Alice repeats the proof process to prove her identity to Bob using Bob's marked authenticator in the subroutine. Finally, Bob can complete the verification process by validating the image created from w^{czd} to the image he received in the second message from Alice, $f(w^{czd})$. Eve's presence shall be known



Alice
Bob
E[r^cd](w^dz′, w^dz)
E[r^cd](w^cz′′, w^cz, f(w^dz′c), f(w^dzc))
proof(q, q^d, w^cz, w^czd)
proof(q, q^c, w^dz, w^czd)

Figure 7.3: Scenario 5

during the proof subroutine in addition to the final image verification check.

7.4 Scenario 6

Scenario six presents an authentication protocol between two parties who have a friend in common without revealing their common friend's identity. Figure 7.4 illustrates the communication exchange between Alice and Bob during the authentication process. The scenario begins where both parties generate a random value that will be used to mask authenticators. Bob initiates the exchange by marking a set of authenticators with his random value before releasing them for Alice. Alice responds with a set of her own marked authenticators. In addition, Alice marks the authenticators she received from Bob and returned an image produced from a one-way function. Bob retains the images for verification.

The next two communication exchanges use the 'proof' subroutine to prove that the given authenticators originated from Alice and Bob, similarly as shown in

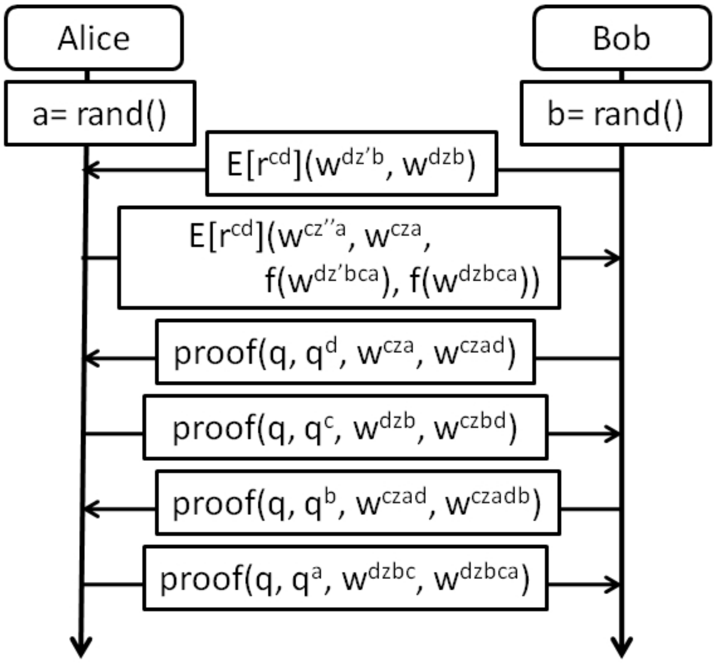

Alice
Bob
a= rand()
b= rand()
$E[r^{cd}](w^{dz'b}, w^{dzb})$
$E[r^{cd}](w^{cz''a}, w^{cza}, f(w^{dz'bca}), f(w^{dzbca}))$
$proof(q, q^{d}, w^{cza}, w^{czad})$
$proof(q, q^{c}, w^{dzb}, w^{czbd})$
$proof(q, q^{b}, w^{czad}, w^{czadb})$
$proof(q, q^{a}, w^{dzbc}, w^{dzbca})$

Figure 7.4: Scenario 6

scenario five. The last two communication exchanges use the 'proof' subroutine to prove that the random values originated from Alice and Bob. Otherwise, the proof subroutine would detect a modification to an authenticator by Eve.

7.5 Scenario 7

Scenario seven outlines a protocol for a friend-of-a-friend communicant discovery system. The scenario begins when Alice and Bob wish to authenticate to each other using a common friend; however, they do not have any friends in common. They could leverage their friends' friends to build a trusting relationship. Scenario seven provides that ideal solution while utilizing concepts from scenario five.

Figure 7.5 outlines the basic communication exchange between all parties in-

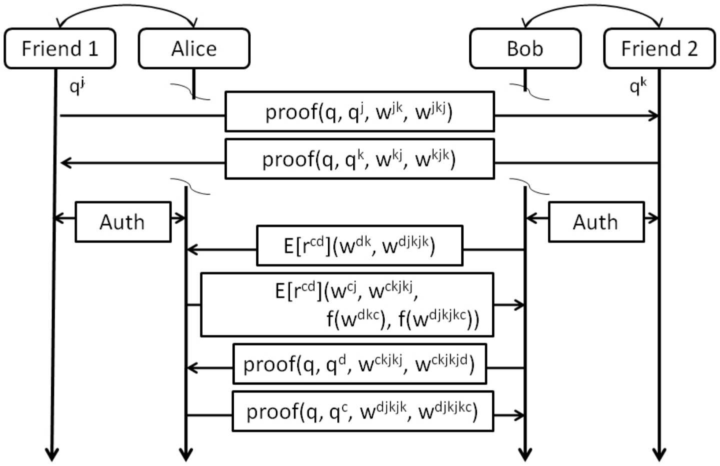

Friend 1
Alice
Bob
Friend 2
q^j
q^k
proof(q, q^j, w^{jk}, w^{jkj})
proof(q, q^k, w^{kj}, w^{kjk})
Auth
Auth
E[r^{cd}](w^{dk}, w^{djkjk})
E[r^{cd}](w^{cj}, w^{ckjkj}, f(w^{dkc}), f(w^{djkjkc}))
proof(q, q^d, w^{ckjkj}, w^{ckjkjd})
proof(q, q^c, w^{djkjk}, w^{djkjkc})

Figure 7.5: Scenario 7

volved. Alice and Bob have a previously established relationship with 'Friend 1' and 'Friend 2,' respectively. Friend 1 and Friend 2 prove that their authenticators they created are 'fresh' and valid as illustrated in the first set of proof transactions. Authenticators may become stale when users change their private and public keys. Next, Alice and Bob must reauthenticate with their corresponding friends to update their set of authenticators. The authentication process is represented by the "Auth" transaction between each party. The "Auth" exchange is merely the protocol defined in scenario 5. A new authenticator is added to Alice and Bob's authenticator set that proves the relationships from Alice to Friend 1 to Friend 2 and Bob to Friend 2 to Friend 1. The remaining exchange between Alice and Bob follows the protocol outlined in scenario five as explicitly defined in Figure 7.5. Eve is caught if any of the proof exchanges fail.

Chapter 8

Discussion

The man-in-the-middle detection protocol determines if a MITM adversary exists for specific scenarios using characteristics of the communication medium that can not be controlled by the adversary. Chaum's patent outlines the MDP and the mechanics of the protocol for each scenario. However, the patent lacks the technical evaluation using conventional notation, the introduction of Eve, and the technical clarity for exchanges in each scenario. While the solution is unique, it is not intuitive. Sections 3, 5, and 6 provide the technical constructs to clearly explain the MDP for each scenario. The next section identifies the contributions added based on this MDP.

A MITM adversary controls the communication link between two parties. The adversary may eavesdrop, modify, block, replay, and slice messages en route without her victim's permission or knowledge. Therefore, it is difficult to define a mechanism to detect an adversary using the same insecure link she controls. Rather, resources must be used that cannot be controlled by an adversary. Researched solutions account for this property using predistributed information, shared secrets, or even a secure authentication server. Each solution implies the assumption that the out-of-band communication is secure.

The MDP follows a unique approach to detect the adversary without needing additional resources or meeting pre-existing conditions. Both communicating parties modify their communication environment based on their derived values of y as an approach to convey the value y itself. The adversary cannot modify or control her victim's expectation of the communication environment since those attributes are never sent across the insecure link. For example, if y influences who should talk first and the duration of the message, then Eve cannot change those characteristics since the y values are never exchanged directly. However, Eve can attempt to match her victim's expectations to avoid detection. The protocol defined in stage three will influence the level of difficulty to maintain a stealthy presence.

To avoid detection, the MITM adversary must match her victim's environmental changes to meet their expectations. With complete control of the communication link, the adversary has the ability to *act* like her victim. However, the adversary cannot *think* like her victims. Thus meaning, Eve cannot generate original messages impersonating either party. The small distinction gives the adversary more control while imposing a less stringent assumption on the MDP. In the event that Eve attempts to replay a previously recorded message, her victims will notice the disruption in the logical conversation flow and raise suspicion.

The MDP scenarios are tailored towards person-to-person communication. For example, a person must detect a disruption in the logical conversation flow in scenario one or verify that a punch-line matches a joke as in scenario two. Human interaction needed within a security enhanced protocol will result in judgment error or confusion. The critical concept embedded in stage two that contains the core components of the MDP does not relay on user decisions. Rather, several scenarios defined in stage three may require user judgment. The processes tailored towards each scenario should minimize or eliminate the user in the decision model that proves a connection is secure since many factors may negatively influence a user's perception such as time, impatience, or fatigue.

Several scenarios were described that outlined how Alice and Bob could convey their derived y values from stage two. These scenarios are examples showing how to modify the characteristics of a communication link to detect an adversary. This technique could be expanded to integrate other scenarios tailored towards their specific goals (i.e. file transfer, voice transmissions, SSL-enabled access, etc...) In addition, new scenarios that follow the core concept of committing to an image before releasing the key provides the ability to collapse stages two and three of the conventional MDP into one stage as illustrated in the friend-of-friend scenarios. The flexible arrangement of the concept allows for expandability and scalability.

Given the ability to adapt the MDP protocol to meet an existing communicating exchange, the MDP patent application claims that vendors have the flexibility to deploy this capability in a software update to new and existing hardware solutions. Given the complex nature of the scenarios from a user's perspective, vendors may find difficulty taking a risk to release the MDP capability that may promote confusion and dissatisfaction experienced by their customers. Before the manufacture community will consider the risk, the MDP must be implemented and supported through another venue as a proof-of-concept.

Overall, the MDP provides a unique solution to detect a MITM adversary using a core concept. However, the protocol lacks the user clarity and usability to gain acceptance in the manufacturer's market. A well defined scenario that eliminates the user's participation to definitively detect a MITM attack may bridge the gap between a prototype and a deployable solution.

Chapter 9

Conclusion

I have described Chaum's MITM detection protocol while providing an illustrative representation of the adversary's engagement in each scenario. I also discussed how the MDP contributed value in the research community by providing a technique that lacks the assumptions of pre-distributed information exchange, shared secrets or the use of an out-of-band secure channel. The most notable difference between the MDP and the detection protocols identified within this paper was described by the assumption that an adversary may act like her victims, but she cannot think like her victims; therefore, each party can detect an active MITM attack since they are cognizant of their peer's communication posture, an attribute the adversary cannot mimic.

Chaum's MDP provides a unique solution; however, public key exchange servers are becoming a predominant component in secure infrastructures as the de-facto standard. Servers that authenticate public keys provide a high level of confidence that two parties are not victims of a MITM attack as compared to the MDP. Furthermore, security that integrates a user decision process may be the largest flaw found within the MDP. As a result, government agencies and other commercial industries would not find it in their best interest to protect their information using the MDP based on a user's decision.

There are still a number of challenges that need to be addressed before the MDP could be seen as a valued capability in consumer products as claimed in the patent. One challenge would be creating a scenario that removes the human component from the detection process. Another challenge would be convincing product manufactures to implement and distribute the MDP capability in their products. Once those challenges are addressed, then it may be feasible to find the MDP as an alternate method to secure communication.

Bibliography

[1] R Atkinson. RFC 1825: Security architecture for the internet protocol. Technical report, 1995.

[2] Gavin Lowe August. An attack on the needham-schroeder public-key authentication protocol. *Information Processing Letters*, 56:131–133, 1995.

[3] Steven M. Bellovin and Michael Merritt. An attack on the interlock protocol when used for authentication. *IEEE Transactions on Information Theory*, 40(1):273–, 1994.

[4] David Chaum. Distributed communication security systems. Patent Application, 2006.

[5] Trevor Perrin. Public key distribution through cryptoids. In *In Proc. Workshop on New Security Paradigms. ACM, 2003*, pages 87–102. [Online] http://trevp.net/cryptoID/cryptoID.pdf, 2000.

[6] Ronald L. Rivest and Adi Shamir. How to expose an eavesdropper. *Commun. ACM*, 27(4):393–394, 1984.

[7] Unknown. Defense against middleperson attacks. `http://web.archive.org/web/20040411015400/http://zooko.com/defense\_against\_middleperson\_attacks.html`, Oct 2003. [Online; accessed 12-January-2010].

[8] Phil Zimmermann. ZRTP: Media path key agreement for secure rtp, Jan 2010.

Notes

[1] Following conventional notation, Alice and Bob shall represent two communicating entities. Eve shall represent the MITM adversary.

[2] The MDP is not limited to the D-H key exchange algorithm. Other public key cryptographic algorithms may be used in exchange.

[3] The term ”proof” was used by the author of the patent for a subroutine found in the friend-of-a-friend scenarios.

[4] ZRTP is a protocol that has not been standardized.