Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

Secure Communication using Identity Based Encryption

Sebastian Roschke[1], Luan Ibraimi[2], Feng Cheng[1], and Christoph Meinel[1]

[1] Hasso Plattner Institute (HPI), University of Potsdam,
P.O.Box 900460, 14440, Potsdam, Germany
{sebastian.roschke, feng.cheng, christoph.meinel}@hpi.uni-potsdam.de
[2] University of Twente,
P.O.Box 217, 7500 AE Enschede, The Netherlands
ibraimi@utwente.nl

Abstract. Secured communication has been widely deployed to guarantee confidentiality and integrity of connections over untrusted networks, e.g., the Internet. Although secure connections are designed to prevent attacks on the connection, they hide attacks inside the channel from being analyzed by Intrusion Detection Systems (IDS). Furthermore, secure connections require a certain key exchange at the initialization phase, which is prone to Man-In-The-Middle (MITM) attacks. In this paper, we present a new method to secure connection which enables Intrusion Detection and overcomes the problem of MITM attacks. We propose to apply Identity Based Encryption (IBE) to secure a communication channel. The key escrow property of IBE is used to recover the decryption key, decrypt network traffic on the fly, and scan for malicious content. As the public key can be generated based on the identity of the connected server and its exchange is not necessary, MITM attacks are not easy to be carried out any more. A prototype of a modified TLS scheme is implemented and proved with a simple client-server application. Based on this prototype, a new IDS sensor is developed to be capable of identifying IBE encrypted secure traffic on the fly. A deployment architecture of the IBE sensor in a company network is proposed. Finally, we show the applicability by a practical experiment and some preliminary performance measurements.

1 Introduction

Secured communication has been widely deployed to guarantee confidentiality and integrity of connections. To achieve this, cryptographic techniques are usually applied to secure the connections, e.g., hash algorithms to secure integrity, asymmetric encryption to secure key exchange, and symmetric encryption to secure confidentiality. Various implementations have emerged and are widely used in practice, such as SSL/TLS [1] and IPSec [2] which provide encryption at the transport layer and the Internet layer. Additionally, a connection can also be secured on the application layer, e.g., simple E-Mail encryption using PGP [3] or XML Encryption. Although secure connections are designed to prevent attacks

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

on the connection, they hide attacks inside the channel from being analyzed by Intrusion Detection Systems (IDS). Furthermore, secure connections require a certain key exchange to work properly. For instance, TLS requires the exchange of a public key between a client and a server, which is prone to Man-In-The-Middle (MITM) attacks [25,26].

Effective prevention of attacks on the communication, e.g., IDS and anti-virus, needs to check the channel, either network traffic or the application layer data on the wire, which leads to several problems. The first problem is performance, as both a network-based IDS sensor and anti-virus software have to analyze large amounts of data in a short time. The second problem is detection in the encrypted traffic. Securing connections by encryption (e.g. SSL/TLS, IPSec, PGP) render the network based IDS and anti-virus software useless, as it is not able to decrypt and to recognize malicious data inside encrypted communication. On the other hand, most of existing cryptography based secure communication require insecure key exchange at the initialization phase. It is not difficult for attackers to exploit this phase, get the keys and carry out further steps of concrete MITM attacks.

In an Identity-Based Encryption[12] (IBE) scheme, the public key of the user is derived from its unique identity, e.g., email address or IP address. The Trusted Authority (TA) or the Key Generation Center (KGC) generates the corresponding private keys. Thus, IBE does not require a digital certificate to certify the public key. Key escrow is an inherent property in IBE systems, i.e., the TA can generate each users' private key, because the TA owns the master key **MK** used to generate users' secret keys. This property can be useful for IDS to decrypt network traffic on the fly and perform content scanning on encrypted traffic.

In this paper, we present a new method to secure connections which provide capabilities for Intrusion Detection and overcome the problem of MITM attacks. We propose to apply IBE to secure communication channels. By integrating the TA in the IDS sensor, it is possible to use the key escrow property for decrypting network traffic on the fly and scan for malicious content. In this way, we can detect malicious content in encrypted network stream without knowing the original private key of the receiver, which can be created on runtime using the master key. The decryption and detection are performed using an IBE sensor. As the public key can be generated based on the identity of the connected server and its exchange is not necessary, the method is not prone to MITM attacks any more. To verify our concept, we implement a prototype of a modified TLS scheme with a simple client-server application. Based on this prototype, we develop a Snort *Preprocessor* to recognize and decrypt IBE secured traffic on the fly. A deployment architecture of the IBE sensor in a company network is proposed. Finally, we show the applicability by a practical experiment and preliminary performance measurements. The contributions of this paper can be summarized as follows:

1. A new mechanism to secure connections, which enables detection of malicious traffic and prevents MITM attacks, is proposed.

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

2. A modified TLS scheme is implemented.
3. A Snort preprocessor is realized to support detection in IBE secured connections.
4. Practical experiments and preliminary measurement results are presented.

The rest of the paper is organized as follows. Section 2 covers the related work, including an overview on secured connections and the IBE encryption scheme. In Section 3, we describe a modified TLS scheme based on IBE. In Section 4, we propose an important application of the IBE scheme for IDS. Section 5 covers the implementation details of the modified TLS as well as the IBE sensor. In Section 6, we shortly describe the experiments based on our prototype implementations and the results of the performance measurements. After some suggestions for future work in Section 7, the work is concluded in Section 8.

2 Related Work

2.1 Secure Network Connections

As the successor of Secure Sockets Layer (SSL), Transport Layer Security (TLS) is a cryptographic protocol that provides integrity, confidentiality, and authenticity for network communications on the transport layer of the TCP/IP protocol stack [1]. It consists of the TLS Record Protocol and the TLS Handshake Protocol. The TLS Record Protocol uses symmetric encryption and a negotiated secret key to secure the connection, e.g., RC4, Triple DES, AES, IDEA, DES, or Camellia. Furthermore, it uses hash functions to guarantee the integrity of a message, e.g., HMAC-MD5 or HMAC-SHA. The TLS Handshake Protocol allows server and client to authenticate each other and to negotiate protocol parameters, such as an encryption algorithm and the cryptographic keys. The identity can be authenticated using asymmetric crypto algorithms, e.g., RSA, DSA, or ECDSA. For key exchange, different algorithms can be used, such as RSA, Diffie-Hellman, ECDH, SRP, or PSK [5].

The TLS Handshake Protocol provides the negotiation of the parameters necessary to establish the secure connection. The used crypto protocols being used as well as the related keys are communicated. During the handshake, at least the server is being authenticated (client authentication is also possible). The TLS handshake starts when a client requests a secure connection from a TLS enabled server. As next step, the client presents a list of supported ciphers and hash functions. The server selects the strongest cipher and hash function available by matching own supported functions and finally negotiates the parameters. Furthermore, the server sends its identification in the form of a digital certificate, e.g., an X.509 certificate, which contains the server name, the trusted certificate authority (CA), and the server's public key K_spub. The client and server exchange the random numbers R_c and R_s to resist replay attacks. After verification of the certificate, the client encrypts a premaster secret PMS with the server's public key K_spub: $E(PMS, K_spub)$ and sends it to the server, which decrypts it with its private key K_spriv: $E(PMS, K_spriv)$. Based on PMS, R_c,

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

and R_s, the server and the client generate the master secret MS, which is used for encryption of the connection $MS = genkey(R_c, R_s, PMS)$. This procedure can be illustrated as follows:

1. Negotiation of encryption and hashing algorithms for client and server and random numbers R_c and R_s
2. Transmission and verification of certificate including K_spub
3. Exchange of R_c and R_s
4. Encryption and transmission of PMS: $E(PMS, K_spub)$
5. Generation of MS based on R_c, R_s, and PMS:$MS = genkey(R_c, R_s, PMS)$

Even if the handshake is sniffed, the confidentiality of the messages is preserved, as the master key MS is impossible to be obtained by the attacker. Thus, attacks inside a secured channel can not be realized. However, the SSL/TLS protocol is prone to MITM attacks [25,26]. The attacker intercepts packets between the client and the server to establish two encrypted connections: 1) a connection between the client and the attacker and 2) a connection between the attacker and the server. The attacker generates its own public key and modifies the original certificate to trick the user/client, which is even possible without a capability for the user to notice the attack [25]. By doing this, the attacker can read the traffic of the secured connection between the client and the server. To get position of a MITM, the attacker can use well known techniques, such as ARP Spoofing, DHCP Fake, DNS Spoofing, or ICMP Redirects [27].

To increase the system performance, the SSL scheme can be supported by SSL acceleration [17]. SSL acceleration is implemented by a hardware crypto device which processes the SSL handshake and in some cases the symmetric decryption. Such devices are optimized for cryptographic operations and offer a very high performance. The crypto device can be installed on the secured server to decrease excessive load on the server. There are also dedicated SSL acceleration appliances which can be placed in the network to perform SSL decryption and encryption. In this case, the SSL accelerator works as a proxy and provides encrypted SSL traffic to the external network, and decrypted traffic to the internal network. There are different accelerator devices available for deployment in a company's network, e.g., Array SPX1800 [18], AppXcel [19], and IBM Crypto Cards [20].

2.2 ID-based Encryption Scheme

The idea of generating users' public key from the identity was proposed by Shamir [11], and the first practical scheme was given by Boneh and Fraklin [12].The Boneh-Franklin scheme consists of four algorithms:

1. $\mathsf{Setup}(k)$: Run by the TA, given a security parameter k, the algorithm generates two cyclic groups $\mathbb{G}$ and $\mathbb{G}_1$ of prime order p, a generator g of $\mathbb{G}$, a bi-linear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$, a master secret key $\mathbf{MK} = \alpha \in \mathbb{Z}_p^*$, and the hash functions $\mathsf{H}_1 : \{0,1\}^* \rightarrow \mathbb{G}$ and $\mathsf{H}_2 : \mathbb{G}_1 \rightarrow \{0,1\}^n$. The publicly available system parameters are $params = (\mathbb{G}, \mathbb{G}_1, p, g, \mathsf{H}_1, \hat{e}, \mathbf{PK}_{TA})$, where $\mathbf{PK}_{TA} = g^\alpha$ is the public key of the TA.

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

2. $\mathsf{Extract}(ID)$: Run by the TA, the algorithm takes as input an identifier ID, and outputs the private key $\mathbf{SK}_{ID} = \mathbf{PK}_{ID}^{\alpha}$, where $\mathbf{PK}_{ID} = \mathsf{H}_1(ID)$.
3. $\mathsf{Encrypt}(m, ID)$: Run by the message sender, the algorithm takes as input the message m and an identifier $ID \in \{0,1\}^*$, and outputs the ciphertext $c = (c_1, c_2)$ where $c_1 = g^r$, $c_2 = m \oplus \mathsf{H}_2(\hat{e}(\mathbf{PK}_{ID}, \mathbf{PK}_{TA})^r)$, where $r \in \mathbb{Z}_p^*$.
4. $\mathsf{Decrypt}(c, sk_{ID})$: Run by the message receiver, the algorithm takes a input the ciphertext $c = (c_1, c_2)$, and the user's secret key $\mathbf{SK}_{ID}$, and outputs the message $m = c_2 \oplus \mathsf{H}_2(\hat{e}(\mathbf{SK}_{ID}, c_1))$.

3 A Modified TLS Protocol for IBE Secured Communication

In IBE scheme, if the TA intercepts the communication between Alice and Bob, the TA can decrypt the encrypted data sent from Alice to Bob, and vice versa. For example, when Alice sends an encrypted email to Bob, Alice, firstly, generates Bob's public key: $\mathbf{PK}_{Bob} = \mathsf{H}_1(bob@mail.com)$, and then sends the encrypted data: $c_1 = g^r$, $c_2 = m \oplus \mathsf{H}_2(\hat{e}(\mathbf{PK}_{Bob}, \mathbf{PK}_{TA})^r)$ to Bob. A Trusted Authority(TA) intercepting the communication between Alice and Bob, can compute $c_2 \oplus \mathsf{H}_2(\hat{e}(\mathbf{PK}_{Bob}^{\alpha}, c_1))$ to reveal the message m. In practice, the message m can be a virus, a trojan horse, a worm, etc, which can not be detected in encrypted form. It is expected to use IBE for securing the email communication so that the TA can be used to decrypt encrypted ciphertext and make it possible to scan for malicious code.

By providing a modified TLS scheme, it would be possible to also use IBE to secure connections on the network. Modification of the TLS specifications and implementation is necessary, as both do not support the IBE scheme. A possible modification can be done by introducing IBE as supported encryption method and modifying the TLS handshake to use IBE during parameter negotiation. Introducing IBE as encryption method is needed as there are no easy ways to deploy encryption methods at runtime or add encryption methods unknown to TLS. Modifying the TLS handshake in case of IBE as encryption method is reasonable as a certificate providing the public key is not needed any more. The public key could be derived from the identity of the server, which needs to be provided. The identity of the server could be the URL or the IP of the system. So the client could directly encrypt the PMS messages with the identity derived key of the server. The scheme looks as follows:

1. Negotiation of algorithms to use and exchange R_c and R_s
2. Generation of IBE public key K_spub based on the identity of the server ID_s: $K_spub = genkey(ID_s)$
3. Encryption of the pre-master-secret PMS: $E(PMS, K_spub)$ with the public key K_spub and transmission to server
4. Generation of MS based on R_c, R_s, and PMS:$MS = genkey(R_c, R_s, PMS)$

The key escrow property is used to achieve IDS capabilities, i.e., decrypting network traffic on the fly and scan for malicious content. In this way, we can detect malicious content in encrypted network stream without knowing the original

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

private key of the receiver, which can be created on runtime using the master key. The decryption and detection are performed using an IBE sensor. As the exchange of the certificate is not needed, the attacker has no chance to forge the certificate and the corresponding public key as a MITM. The client will generate the public from the identity of the server and will directly encrypt the session key with this public key. The attacker can intercept this message, but he is not capable of reading the encrypted session key. Thus, a MITM attack to establish two secured channels that appear as one to client and another one to the server can not be performed. He still can disturb the connection, but the confidentiality and integrity of the communication are secured. The only requirement is that the parameters of the cryptosystem are known before. This can be achieved by a similar method as the distribution of trusted certificates for TLS.

4 Detecting Intrusions in the modified TLS Communication

The modified TLS protocol makes it possible to detect intrusions in encrypted communication channels. To realize this, the IBE supported IDS sensor (which is called IBE Sensor) is required. We apply the proposed modified TLS in some normal services deployed in a typical enterprise network with a demilitarized zone (DMZ) and the internal subnet. In the internal network, there are several connected clients and the PKI infrastructure for IBE. Within the DMZ there are several servers running, providing services for internal and external clients. To support IBE, there are private keys SKP_i deployed on all servers and CKP_i deployed on clients in the network. These private keys are derived from the master key **MK**. The deployed IBE sensor should be able to read all incoming and outgoing network traffic in the DMZ as well as traffic in the internal network.

Using the master key **MK** it would be possible to decrypt the traffic which is encrypted based on the IBE scheme. If the modified TLS scheme is used, the runtime decryption works by extracting the master secret from the TLS handshake while sniffing all network traffic. This is achieved using the following steps:

1. Recognize negotiation of algorithms and exchange of R_c and R_s (store R_c and R_s)
2. Recognize encrypted pre-master secret $E(PMS, K_spub)$ on the network and decrypt it with master key **MK**: $D(E(PMS, K_spub), \mathbf{MK}) = PMS$ (store PMS)
3. Generate MS based on R_c, R_s, and PMS:$MS = genkey(R_c, R_s, PMS)$
4. Use MS to decrypt the encrypted traffic

As this method is passive, no server or client would be disturbed during its communication. The IBE sensor does not need to intercept and forward traffic, but it needs to store information about the TLS session to assign network traffic to a specific connection/TLS session and a related master secret MS.

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

The distribution of the master key **MK** yields additional risks for the network. As the master key **MK** is able to decrypt any traffic within the network, it might be a main target for attackers. While the risk for a PKI is known and confined by security mechanisms, the possible risks for an IBE sensor are higher as it is exposed to any traffic on the network. Furthermore, the IBE sensor relies on traffic parsers which might be vulnerable to different kinds of attacks, e.g., Buffer Overflows or Format String Exploits [6]. To secure the IBE sensor and the master key **MK**, additional security measures need to be realized. By successfully attacking the IBE sensor, the attacker will be able to passively sniff all encrypted traffic on the network. With knowledge on the identities, the attacker will be able to generate the private keys and to send falsified messages with a spoofed identity.

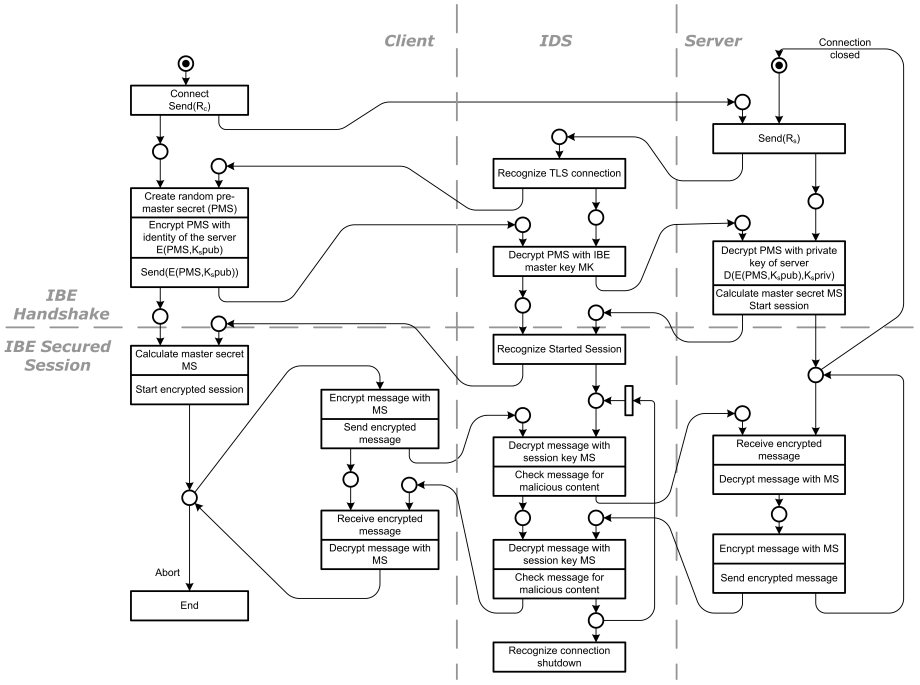

Fig. 1. IBE Sensor Secured Client-Server Communication

5 Implementation

The proof-of-concept implementation consists of two parts: the implementation of a modified TLS specification and the implementation of an IBE-supported Snort IDS sensor (IBE sensor). The modified TLS is implemented between a

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

client and a server. The communication between the client and the server is encrypted based on a session key, which is securely exchanged during the session negotiation by using the IBE encryption scheme. The Snort IDS sensor is modified to recognize this session negotiation and restore the session key based on the key escrow property.

Figure 1 shows the communication between the client and the server secured by the modified TLS scheme. The scheme consists of two separated phases: the session negotiation and the encrypted session. The initial state is a server listening for connections. After the client connects to the server, it receives the banner and creates a random session key *key_iv*. Then it encrypts the *key_iv* with the identity of the server based on IBE and sends it to the server. The server can decrypt the session key with its private key based on IBE and acknowledge that the session is established. In the following, the client sends multiple encrypted messages to the server. The session is encrypted based on the symmetric Blowfish cipher[23].

An IDS sensor consists of a detection engine, a reporting engine, and a comprehensive rule set for detection. The Snort IDS sensor provides several extension possibilities realized by so called *Dynamic Modules* [4]. *Dynamic modules* can be additional detection rules for the existing detection engine, a new detection engine, or so called *Preprocessors*. A *Preprocessor* is a configurable software component in the Snort architecture, which is called before the detection engine. After the packet has been processed by all preprocessors, it is given to the detection engine to be scanned in detail according to the configured rule sets. To decrypt the TCP stream on the fly before detection, we implemented a *Preprocessor*, which is capable to recognize the modified TLS scheme in a TCP stream.

As shown in Figure 1, the modified Snort can be used to detect attacks in a TCP stream which is encrypted by the modified TLS scheme. To identify a new session, the IDS recognizes the session negotiation, which is sent by the server. The next information sent by the client is the encrypted pre-master secret *PMS*, which can be decrypted by IDS using the master key. By using the extracted *PMS*, the IDS can calculate the session key as well as decrypt and analyze all the network traffic in this session. For this purpose, the *Preprocessor* gets all packets in this session, decrypts it, and modifies the packet on runtime to make it being analyzed by the detection engine. The implementation is done in plain C with support of the OpenSSL library for symmetric encryption and decryption. The session negotiation is implemented on the client and server based on the Stanford IBE implementation [24].

6 Experiment Results

In the experiment scenario, there are several clients and an attacker host connected to the external network, e.g., the Internet. The internal network is attached to the external network using a firewall. The IBE sensor is configured to monitor the internal network and to hold the master key, which is used to gener-

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

ate the server keys SKP_1 to SKP_m. In the internal network there is a server to hold SKP_m and has the identity "server@hpi". The clients can connect to this server using the modified TLS scheme, which secures the communication session based on IBE and Blowfish. The sensor is running with an *Intel(R) Core(TM)2 Duo* at $1.4GHz$, i.e., it includes two dedicated cores with $3.072KB$ of cache each. The system has $2GB$ of *RAM*.

During the experiment, we have several clients sending encrypted messages to the server based on the modified TLS scheme. Since he/she can pretend to be a normal user, the attacker is also using the modified TLS scheme to establish a session with the server. Additionally, the attacker tries to send malicious content (e.g., x86 shellcode) to the server within this encrypted session. As the traffic is encrypted based on a session key, the unmodified Snort sensor can not read it and therefore can not detect the shellcode. By using the IBE sensor, which recognizes the session negotiation, we can detect the shellcode within the encrypted session.

The attacker sends shellcode to the server within the encrypted session. This shellcode could be used to exploit the server without being detected by an IDS sensor. The IBE sensor decrypts the session and finds the malicious shellcode in the network traffic. The output is the Snort alert file, which contains the detected attacks on the wire. We also tested other content-based detection rules with the prototype, such as the *Backdoor Rules* used to detect backdoor communication on the network or the *Attack Response Rules* used to detect special responses given by successful attacks [4]. All the tested rule sets worked fine for our prototype.

6.1 Performance Analysis

To test the performance of our implementation, we measured the processing time in the Snort implementation while sending 10 messages with different size from the client to the server. To compare the results, we distinguished three different scenarios. The scenarios are chosen to give a first impression on the performance of the approach. The message sizes in the real world depend on the actual application scenarios:

1. Small size message content - 10 messages, each with 20 Bytes
2. Medium size message content - 10 messages, each with 80 Bytes
3. Large size message content - 10 messages, each with 480 Bytes

We analyzed the results for the key negotiation and the encrypted session separately, to avoid confusion. Finally, Table 1 shows the average values of the processing time comparing the different scenarios, the key negotiation and encrypted session, as well as the processing of Snort with and without runtime decryption. The average of packets during the key negotiation is similar across different scenarios. While the processing time of the packets without runtime decryption is around 35 micro second, the processing time with runtime decryption is around 3600 micro seconds. This is reasonable as the IBE decryption process during the key negotiation needs a lot of additional effort. Furthermore,

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

the effort is the same for the three different scenarios, as the messages which are exchanged during the key negotiation are basically the same. Comparing the processing times of the packets in an encrypted session, one can realize that the average without runtime decryption is around 28 micro seconds, while the processing time with runtime decryption depends on the message size. The experiment shows the significantly higher processing time of the scenario with large size message content, i.e., 480 Bytes per message. Each encrypted packet needs between 110 and 1250 micro seconds to be processed. The processing time of the scenarios with small size and medium size messages is similar, i.e., between 180 and 210 micro seconds. This can be explained by the functionality of the Blowfish cipher, which is a block cipher. A clear text message is padded to a specific block size (or a multiple of it) to be encrypted with this cipher. The default block size of the Blowfish cipher is 64 bits while our encryption uses 128 bits cleartext blocks to be encrypted. The message size of the two scenarios is padded to the same block size, which needs the same amount of processing time. Therefore, the processing time for both scenarios appears to be the same.

	Scenario	Snort Process() without Decryption	Snort Process() with Decryption
Key Negotiation	small	36,5	3820,2
	medium	34,7	3351,75
	large	35,6	3616,88
Encrypted Session	small	30,65	76,29
	medium	29,4	75,73
	large	25,52	369,65

Table 1. Performance Table: Average Snort processing time in micro seconds

7 Future Work

As the concept shows promising results, the integration of IBE into the real TLS standard should be considered in the future. Besides the evaluation of the applicability, it includes the modification of supported and well-known implementations, such as OpenSSL. In addition to our preliminary measurement results, the performance needs to be analyzed in detail and the scheme can be optimized accordingly to ensure scalability. A possible approach to improve performance can be the utilization of hardware devices to support cryptographic operations for the IDS, e.g., the expensive pairing operations and the symmetric decryption. This could improve the performance and make our approach applicable for huge network environments. The described attack countermeasures does not work with Snort in *In-line* mode [4], a specific operation mode where Snort is not separately deployed in the network. In this way, the sensor can drop malicious packets directly while the sensor itself is exposed to the network and to potential attackers. In this case, further security mechanisms are necessary to secure the master key as well as the sensor itself from being compromised.

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

Moreover, it could be interesting to analyze the utilization of Attribute Based Encryption (ABE)[13] for secured connections. ABE is a generalization of IBE, and differs from IBE such that the user in ABE get a secret key associated with a set of attributes, while in IBE the user gets a secret key associated with an identity. Same as IBE, the ABE key escrow is an inherent property since the master key is used to generate each user secret key. A suitable ABE scheme for being applied in IDS is Key-Policy Attribute-Based Encryption (KP-ABE)[15], since it is related to the concept of searching on encrypted data, in which a server on behalf of the user can make a query on the encrypted data without knowing the query and without knowing the plaintext.

8 Conclusion

In this paper, we proposed a new mechanism to create secured connection based on IBE. By using the key escrow property, the secured connection can be recovered and the network traffic is possible to be analyzed by IDS. In this way, we can detect malicious content in encrypted network streams without knowing the original private key of the receiver. Additionally, the secure connections based on IBE have no need to exchange public keys in certificates, which prevents MITM attacks on the secured channel. We provide a deployment architecture of the IBE sensor in a company network. Furthermore, we implemented a prototype of a modified TLS scheme with a simple client-server application to support IBE. Based on this prototype, we developed an IBE sensor (i.e., Snort *Preprocessor*) to recognize and decrypt IBE secured traffic on the fly. Finally, we showed the applicability by a practical experiment and analyzed the performance of our implementation.

References

1. "The TLS Protocol" Website: http://www.ietf.org/rfc/rfc2246.txt (accessed Jan 2010).
2. "Security Architecture for the Internet Protocol" Website: http://www.rfc-editor.org/rfc/rfc4301.txt (accessed Jan 2010).
3. "OpenPGP Message Format" Website: http://tools.ietf.org/html/rfc4880 (accessed Jan 2010).
4. "Snort IDS" Website: http://www.snort.org/ (accessed Jan 2010).
5. A. J. Menezes, P. C. van Oorschot, S. A. Vanstone: "Handbook of Applied Cryptography", CRC Press, 1 edition (1996).
6. J. Koziol, D. Litchfield, D. Aitel, C. Anley, S. Eren, N. Mehta, R. Hassell: "Shellcoders Handbook", Wiley Publishing, Inc. (2004).
7. O. Hallaraker and G. Vigna: "Detecting malicious javascript code in mozilla", In: Proceedings of *International Conference on Engineering of Complex Computer Systems (ICECCS'05)*, Shanghai, China, IEEE Computer Society, pp. 85-94 (2005).
8. M. V. Mahoney and P. K. Chan: "An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection", In: Proceedings of *The Sixth International Symposium on Recent Advances in Intrusion Detection (RAID'03)*, Pittsburgh, PA, USA, Springer LNCS, Vol. 2820, pp. 220-237 (2003).

Sebastian Roschke, Luan Ibraimi, Feng Cheng, Christoph Meinel: "Secure Communication Using Identity Based Encryption"
in Proceedings of the 11th IFIP Conference on Communications and Multimedia Security (CMS 2010) , LNCS 6109, Springer, Linz, Austria, pp. 256-267, 5, 2010. ISBN: 978-3-642-13240-7.

9. M. Ramadas, S. Ostermann, and B. C. Tjaden: "Detecting anomalous network traffic with self-organizing maps", In: Proceedings of *The Sixth International Symposium on Recent Advances in Intrusion Detection (RAID'03)*, Pittsburgh, PA, USA, Springer LNCS, Vol. 2820, pp. 36-54 (2003).
10. S. Northcutt: *Network Intrusion Detection - An Analyst's Handbook*, New Riders, (1999).
11. A. Shamir "Identity-based cryptography and signature schemes", In: Proceedings of *Advances in Cryptology (CRYPTO'84)*, Santa Barbara, California, USA, Springer LNCS, Vol. 196, pp. 47-53 (1985).
12. M. K. F. Dan Boneh: "Identity-based encryption from the weil pairing", In: Proceedings of *The 21st Annual International Cryptology Conference on Advances in Cryptology CRYPTO'01*, Santa Barbara, California, USA, Springer LNCS, Vol. 2139, pp. 213-229 (2001).
13. Sahai, A. and Waters, B.: "Fuzzy identity-based encryption", In: Proceedings of Advances in Cryptology (Eurocrypt'05), Aarhus, Denmark, Springer LNCS, Vol. 3494, pp. 457-473 (2005).
14. Bethencourt, J. and Sahai, A. and Waters, B.: "Ciphertext-policy attribute-based encryption", In: Proceedings of the *2007 IEEE Symposium on Security and Privacy (S&P'07)* Washington, DC, USA, IEEE Press, pp. 321-334 (2007).
15. Goyal, V. and Pandey, O. and Sahai, A. and Waters, B.: "Attribute-based encryption for fine-grained access control of encrypted data", In: Proceedings of the *13th ACM Conference on Computer and Communications Security (CCS'06)*, New York, NY, USA, ACM Press, pp. 89-98 (2006).
16. "Voltage security", Website: http://www.voltage.com/ (accessed Jan 2010).
17. "SSL Acceleration", Website: http://sslacceleration.info/ (accessed Jan 2010).
18. "Array Networks: Universal Access Controllers", Website: http://www.arraynetworks.net/entry.asp?PageID=110 (accessed Sept 2009).
19. "Radware: AppXcel", Website: http://www.radware.com/ (accessed Jan 2010).
20. "IBM: Crypto Card", Website: http://www-03.ibm.com/security/cryptocards/ (accessed Jan 2010).
21. B. Irwin: "Unlocking the armour : enabling intrusion detection and analysis of encrypted traffic streams", In: Proceedings of *New Knowledge Today Conference (ISSA KTC'05)*, Sandton, South Africa, ISSA Press, pp. 1-10 (2005).
22. A. Yamada, Y. Miyake, K. Takemori, A. Studer, and A. Perrig: "Intrusion detection for encrypted web accesses", In: Workshop Proceedings of *Advanced Information Networking and Applications (AINA'07)*, Niagara Falls, Ontario, Canada, IEEE Press, pp.569-576 (2007).
23. B. Schneier: "Description of a new variable-length key, 64-bit block cipher (blowfish)", In: Proceedings of *Fast Software Encryption (FSE'93)*, Cambridge Security Workshop Cambridge, UK, Springer LNCS, vol. 809, pp. 191-204, 1993.
24. B. Lynn: "Stanford IBE Library v0.7.2.", Website: http://crypto.stanford.edu/ibe/ (accessed Jan 2010).
25. Stevens, M., Sotirov, A., Appelbaum, J., et. al.: "Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate", In: Proceedings of *29th Annual international Cryptology Conference on Advances in Cryptology (CRYPTO'09), Santa Barbara, CA, Springer LNCS, Vol. 5677, pp. 55-69 (2009).*
26. *Boneh, D., Inguva, S., Baker, I.: "SSL Man in the Middle Proxy ", Website: http://crypto.stanford.edu/ssl-mitm/ (accessed Jan 2010).*
27. *Ettercap: Website: http://ettercap.sf.net/ (accessed Jan 2010).*