SERVER NOTARIES: A COMPLEMENTARY APPROACH TO THE WEB PKI TRUST MODEL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE YÜCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CRYPTOGRAPHY

JANUARY 2016

Approval of the thesis:

SERVER NOTARIES: A COMPLEMENTARY APPROACH TO THE WEB PKI TRUST MODEL

submitted by **EMRE YÜCE** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** _____

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** _____

Assoc. Prof. Dr. Ali Doğanaksoy
Supervisor, **Mathematics, METU** _____

Prof. Dr. Ali Aydın Selçuk
Co-supervisor, **Computer Engineering, TOBB ETÜ** _____

Examining Committee Members:

Prof. Dr. Ferruh Özbudak
Mathematics, METU _____

Assoc. Prof. Dr. Ali Doğanaksoy
Mathematics, METU _____

Assoc. Prof. Dr. Çetin Ürtiş
Mathematics, TOBB ETÜ _____

Prof. Dr. Kemal Bıçakçı
Computer Engineering, TOBB ETÜ _____

Assoc. Prof. Dr. Murat Cenk
Cryptography, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: EMRE YÜCE

Signature :

ABSTRACT

SERVER NOTARIES: A COMPLEMENTARY APPROACH TO THE WEB PKI TRUST MODEL

Yüce, Emre

Ph.D., Department of Cryptography

Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy

Co-Supervisor : Prof. Dr. Ali Aydın Selçuk

January 2016, 49 pages

SSL/TLS is the de facto protocol for providing secure communication over the Internet. It relies on the Web PKI model for authentication and secure key exchange. Despite its relatively successful past, the number of Web PKI incidents observed have increased recently. These incidents revealed the risks of forged certificates issued by certificate authorities without the consent of the domain owners. Several solutions have been proposed to solve this problem, but no solution has yet received widespread adaption due to complexity and deployability issues. In this work, we propose a practical mechanism that enables servers to get their certificate views across the Internet, making detection of a certificate substitution attack possible. The origin of the certificate substitution attack can also be located by this mechanism. We have conducted simulation experiments and evaluated our proposal using publicly available, real-world BGP data. We have obtained promising results on the AS-level Internet topology.

Keywords : Web PKI, SSL/TLS, man-in-the-middle attack, notary

ÖZ

SUNUCU NOTERLERİ: WEB AAA GÜVEN MODELİ İÇİN TAMAMLAYICI BİR YAKLAŞIM

Yüce, Emre

Doktora, Kriptografi Bölümü

Tez Yöneticisi : Doç. Dr. Ali Doğanaksoy

Ortak Tez Yöneticisi : Prof. Dr. Ali Aydın Selçuk

Ocak 2016, 49 sayfa

SSL/TLS Internet'te güvenli iletişim sağlamak için varsayılan protokoldür. Doğrulama ve güvenli anahtar değişimi için Web AAA modeli üzerine kurulmuştur. Başarılı geçmişine rağmen, gözlenen Web AAA olayları son zamanlarda artmıştır. Bu olaylar, alan adı sahiplerinin isteği dışında sertifika otoriteleri tarafından üretilen sahte sertifikaların yarattığı riskleri belirgin hale getirmiştir. Bu problemi çözmek için çeşitli çözümler önerilmiş olmasına rağmen, karmaşıklık ve kurulum sıkıntıları yüzünden hiçbir çözüm yaygın kullanılır hale gelememiştir. Bu çalışmada sunucuların kendi sertifikalarının Internet'teki görünümlerini elde edebilecekleri ve sertifika değiştirme saldırılarını tespit edebilecekleri pratik bir yöntem önermekteyiz. Ayrıca bu yöntem ile sertifika değiştirme saldırılarının kaynağını tespit etmek de mümkün olacaktır. Bu kapsamda kamuya açık gerçek BGP verisini kullanarak önerdiğimiz yöntemi değerlendirmek amacıyla bazı simulasyonlar gerçekleştirdik. Yaptığımız simulasyonlarda AS seviyesinde kayda değer sonuçlar elde ettik.

Anahtar Kelimeler : Web AAA, SSL/TLS, ortadaki adam saldırısı, noter

To My Family

ACKNOWLEDGMENTS

I would like to express profound gratitude to my advisor, Assoc. Prof. Dr. Ali Doğanaksoy for his support and encouragement throughout my undergraduate and graduate education.

My co-advisor, Prof. Dr. Ali Aydın Selçuk, has been always there to listen and give advice. I am deeply grateful to him for pushing me reach my limits. I am also thankful to him for his continuous guidance and support which enabled me to complete my work successfully.

I would also like to thank my committee members, Prof. Dr. Ferruh Özbudak, Prof. Dr. Kemal Bıçakçı, Assoc. Prof. Dr. Çetin Ürtiş, and Assoc. Prof. Dr. Murat Cenk for serving as my committee members.

I am also highly thankful to my colleagues Onur Bektaş and Uğur Yılmaz from TÜBİTAK ULAKBİM for their comments and feedback through this work.

Many friends have helped me stay sane through these difficult years. I cannot forget my friends who went through hard times together, cheered me on, and celebrated each accomplishment. Their support and care helped me overcome setbacks and stay focused on my graduate study. I greatly value their friendship and I deeply appreciate their belief in me. I would like to thank to Neşe Koçak and Onur Koçak for supporting me and believing in me throughout this journey. I also thank to my friends Alper Atalay and Fehmi Par Bekçi for their moral support throughout my entire life.

Most importantly, none of this would have been possible without the love and patience of my family. My family to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to my family. I am as ever especially indebted to my mother, my brother and my father for their love and support throughout my life. I also thank to my beloved wife Canay Aykol Yüce for her continuous support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Today the Internet is massively used for e-government, e-commerce, and e-banking applications unlike its early days with static web pages. These applications require exchange of sensitive data including financial or personal information. It is crucial to provide a secure connection for this communication which is achieved using different network protocols.

Cryptographic primitives confidentiality, authenticity, and integrity should be provided for a secure communication. Confidentiality is protecting data and not making available or not disclosing data to unauthorized entities [1]. Encryption algorithms are deployed to keep curious eyes out from the data content. Integrity is defined as protecting accuracy and completeness of the data and the assurance that the data is not modified along its path to destination by unauthorized entities. Authentication is a process that is used to confirm that a claimed characteristic of an entity is actually correct.

1.1 Symmetric Key Encryption

An encryption scheme is defined as *symmetric key* if calculating the decryption key d using the encryption key e is computationally easy and vice versa. Most of the symmetric key encryption schemes uses $d = e$ as the name implies. It is obvious that the participating entities should keep both d and e secret.

Symmetric key encryption schemes are efficient at processing high volumes of data. However they have integrity and authentication issues. They should be accompanied by message authentication codes (MACs) to guarantee that a message is delivered unchanged. A recipient is not able to authenticate the sender in a symmetric encryption scheme.

It is assumed that the participating entities know a shared secret in advance. This secret may be the decryption/encryption keys or a parameter to generate them. Hence participants should find an efficient way to exchange this secret. This issue is referred as the *key distribution problem* [44].

1.2 Public Key Encryption

Public key encryption schemes use different key pairs for encryption and decryption. Encryption keys are called public keys and the receiver announces his public key so that anyone can send an encrypted message to him. Apart from symmetric key encryption schemes, computing the decryption key using the encryption key is not feasible. This is only possible using an extra information which is kept secret. Public key encryption schemes uses hard mathematical problems and one-way trapdoor functions to achieve this.

Diffie and Hellman introduced the concept of public key cryptography in 1976 [14]. They proposed the Diffie-Hellman key exchange protocol whose security is based on the intractability of the discrete logarithm problem. In 1978 Rivest, Shamir and Adleman has proposed the RSA public key encryption scheme [49] which is based on the integer factorization problem.

Public key encryption schemes has made a remarkable contribution by introducing the digital signatures. The sender binds its entity into the message so that the receiver is able to authenticate the sender. Additionally this makes non-repudiation and authorization possible.

1.3 Active/Passive Adversary Threat Model

An adversary is called a *passive adversary* if it can only observe the messages through a communication but cannot tamper with them. If the adversary is also able to manipulate the messages, it is called an *active adversary*. An active adversary is able to launch a *man in the middle (MITM) attack* by changing the messages in transit.

Symmetric key encryption schemes do not propose a secure key distribution method against either passive or active adversaries. This constitutes a problem in large scale environments like the Internet.

Public key encryption schemes resolve the key distribution problem against passive adversaries. However, if there exist an active adversary between the sender and the receiver, the adversary will be able to impersonate the participating entities to each other.

Consider Diffie-Hellman (DH) key exchange protocol. In a regular DH key exchange, the participating entities Alice and Bob initiate the protocol by choosing publicly known g and p values. Then Alice generates her secret a, calculates $A = g^a \ mod \ p$, and sends A to Bob. Similarly, Bob generates his secret b, calculates $B = g^b \ mod \ p$, and sends B to Alice. Alice calculates $B^a \ mod \ p$ which equals to $(g^b)^a \ mod \ p$. Bob finds the same value by calculating $A^b \ mod \ p$ which equals to $(g^a)^b \ mod \ p$. Hence both participants obtain the same value, i.e. the secret share, which is a natural candidate to be used as the seed in the key generation process.

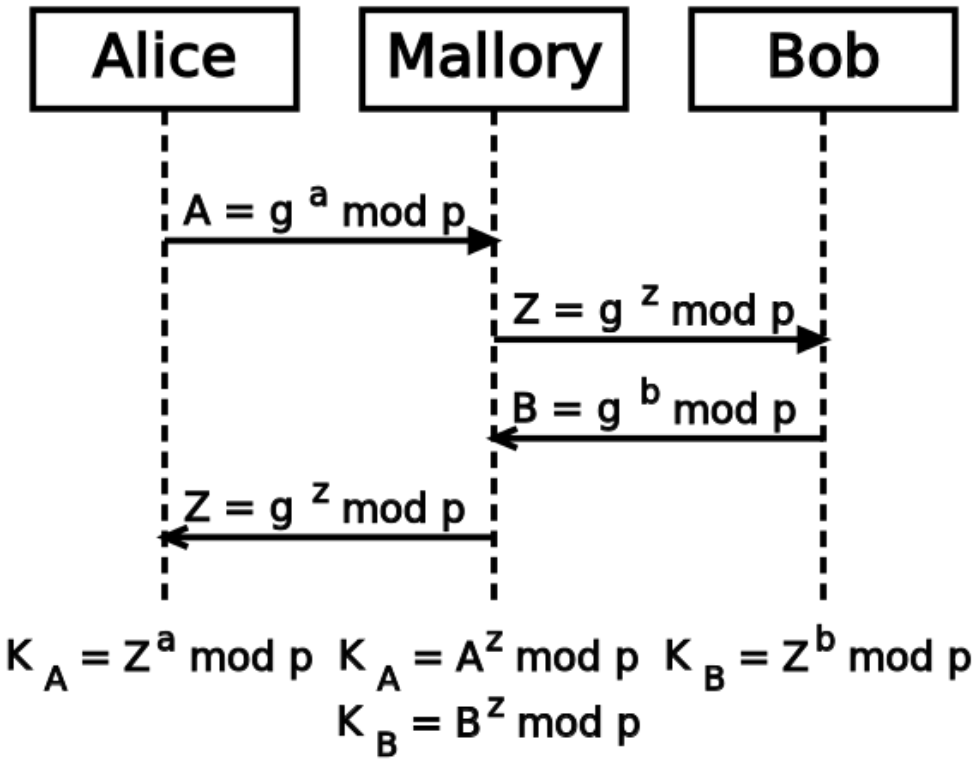

Figure 1.1: Mallory (*the active adversary*) launches a MITM attack against the DH key exchange between *Alice* and *Bob*. Mallory impersonates Alice to Bob and vice versa by interrupting the traffic and injecting its own parameters [5].

DH is a secure key exchange protocol against a passive adversary. However, an active adversary is able to manipulate the secret share by interrupting the messages in transit as shown in Figure 1.1. Mallory observes the messages and receives $A = g^a \ mod \ p$ from Alice and $B = g^b \ mod \ p$ from Bob. Then she interrupts the messages in transit and sends $Z = g^z \ mod \ p$ to both Alice and Bob. Alice calculates $K_A = Z^a \ mod \ p$ and Bob calculates $K_B = Z^b \ mod \ p$. Thus Alice and Bob think they generate the same secret share with each other, however in fact they did it with Mallory. Hence Mallory is able to obtain the keys generated using this secret share.

1.4 Trust Models for Public Key Encryption

A public key infrastructure (PKI) is a framework which combines various components (public key encryption schemes, digital signatures, trusted third parties etc.) by defining necessary procedures (e.g. key management, revocation) to build a trust model. The main issue while constructing a PKI is the decision about what the trust would be based on. There exist models deploying trusted third parties or utilizing already known entities as bootstrapping nodes for trust.

X.509 standard for public key infrastructures was introduced in 1988 [11]. It specifies standards for digital certificates, certificate revocation lists, and a certification path validation algorithm. It introduces digital certificates as the digital documents which bind an entity's public key to its identity information such as a DNS record or an e-mail address.

X.509 standard uses a hierarchical system where only certificate authorities (CAs) are allowed to issue digital certificates for entities. If an entity has the CA's public key, it is able to verify the signature of the CA embedded in a digital certificate, hence the integrity of the digital certificate and the identity of the sender. Thus the trust is anchored to the CAs.

Web of trust is a decentralized PKI trust model unlike the centralized structure of X.509 standard. An entity generates a public/private key pair and associate it with a username and email address. Entities exchange and sign each others' public keys after being sure about the identity (e.g. face to face meeting). For this purpose key signing parties are organized.

In the web of trust model, the user should find a trust chain to the entity he wants to communicate with. In that sense it is not feasible to be used in a large scale environment. Another critical point is that the mapping between the key pairs and the identities is not bijective. A user may have more than one key pair (e.g. using different email addresses) and it is not clear how to choose the real identity.

The trust on first use (TOFU) model is not an actual infrastructure but a kind of procedure. In this model the user assumes that the initial cleartext connection to the entity is not manipulated. Thus the public key of the entity received at the initial connection is trusted by default. A change in the public key will be detected in consecutive connections. This method is also known as *leap of faith*, *pinning* or *key continuity*.

SSH, the de facto remote secure shell access protocol, is an example where TOFU model is used as an alternative authentication method. SSH is used to establish a communication between a server and a client. The proper usage is to introduce the server's public key to the client prior to the connection out-of-band, i.e. by storing the server's public key on the client as a trusted host. Otherwise the client is not able to verify the server's public key. Hence it presents the fingerprint of the received public key to the user and leaves the decision to the user. If the user accepts the public key, it is added to the trusted hosts list for future connections just like the predistributed case. In practice system administrators are prone to blindly accept the fingerprint and establish the first connection using TOFU model.

1.5 Web PKI

Secure Socket Layer (SSL) [21] and its successor Transport Layer Security (TLS) [13] are de facto protocols providing confidentiality, authenticity, and integrity over the

Internet. SSL[1] relies on the *Web PKI* trust model [11] for authentication and secure key exchange.

Web PKI trust model is a framework based on the X.509 standard. In this model, Certificate Authorities (CAs) issue X.509 digital certificates that bind the SSL server identity (e.g. domain name) to a public key. SSL clients receive the digital certificate when they request to establish a secure connection to the server. They verify it using the embedded public keys of CAs in their browser or operating system certificate trust stores. SSL can be used for file transfers, instant messaging, or sending e-mails however the most popular usage is to establish a secure communication between the client and a web server.

There exist serious concerns regarding the reliability of the Web PKI trust model. The model employs a list of CAs that are trusted by default. There are hundreds of fully trusted root CAs from more than 50 countries [17]. They are able to delegate their authority to subordinate CAs (sub-CAs) as well. For any domain name both root CAs and sub-CAs are able to issue valid certificates, trusted by most of the browsers, without the consent or knowledge of the domain name owner.

Another issue is the lack of trust agility as Moxie Marlinspike stated [42]. Current trust model does not allow users to select whom and for how long to trust. A user may choose not to trust to a CA by removing its root certificate from his trust store. However this would result in breaking the trust to any certificate issued by that CA.

1.5.1 Incidents

One of the most recent incidents has happened in September 2015 [23]. Google detected that Symantec's Thawte-branded CA issued an Extended Validation (EV) precertificate for the domains google.com and www.google.com without neither request nor authorization. The incident got more interesting as Google investigated the issue [25]. Finally, Google resolved the issue by removing and distrusting the respective Symantec root certificate [24].

Another recent incident has occurred in March 2015 [36]. Similarly, Google detected forged certificates for several Google domains. A sub-CA certificate, signed by National Informatics Centre of China (CNNIC), has been used in the incident. Browser and operating system vendors revoked the certificates after the discovery of the attack. This attack is an example of misuse of sub-CA certificates.

Other examples are IndiaNIC case in July 2014 [35], ANSSI case in December 2013 [34], and TurkTrust case in January 2013 [33]. Yet in other incidents, CAs were compromised resulting in the fraudulent issue of forged certificates [58, 10]. Governmental and private organizations may also use forged certificates for their surveillance activities [38, 51, 54].

[1] Hereafter, we use SSL to mean both SSL and TLS.

1.6 Proposals to Solve the Web PKI Issues

There exist several proposals suggesting improvements to the current Web PKI trust model. Some of them try to replace the CA infrastructure completely, while others try to fit in and enhance the current model.

1.6.1 DNS-Based Solutions

DNS-based Authentication of Named Entities (DANE) [50] proposes binding SSL keys to DNS entries using DNSSEC. This proposal may be seen as pinning keys to the DNS entries. In order for the DANE solution to be used, the vast majority of DNS servers should be configured to use DNSSEC. Also revocation is again problematic in DANE since all DNS records, including caches, worldwide should be updated in case of a public key update. This depends on the TTL value of the records.

1.6.2 Notary-Based Solutions

The idea of observing the server certificate from different network vantage points has been used in several proposals to improve the Web PKI trust model. This idea was introduced in Perspectives [61], where Wendlandt et al. defined *notaries* as publicly available semi-trusted hosts deployed at various locations on the network. The main idea is that after a client obtains the server certificate in the usual way, it may compare the received certificate with the server certificate obtained from a notary's network point of view. A difference between the certificates may indicate a certificate substitution. Different variants of notaries have been used in several different protocols

Perspectives proposal introduced the notary approach. In this proposal notaries periodically probe network services to build a public key database. If a client receives an unauthenticated public key from a service, it contacts to a notary and downloads the history of observed public keys for that service. Thus the client is able to decide whether the public key is genuine.

Perspectives provides three main protection mechanisms. Firstly, public key observations from multiple vantage points, *spatial* redundancy, enables the client to receive the genuine public key, unless an attacker compromises all network paths to the server. Secondly, public key observations over time, *temporal* redundancy, allows client to observe the key history and detect a recent key change. Finally, *data* redundancy is a mechanism to detect the misbehaving notaries.

In 2011, Moxie Marlinspike proposed another notary-based solution, Convergence [43], which introduces some enhancements to the Perspectives method. Convergence uses bounce notaries to prevent privacy issues. Notaries use other methods (DANE, CAs, etc.) to authenticate the keys.

Doublecheck [2] and DetecTor [12] are similar notary-based proposals which use the

TOR network as the notary infrastructure.

An interesting idea for both detecting and locating the adversaries using notaries, originally called hunters, has arisen in the CrossBear proposal [28].

The ICSI Certificate Notary [30] and the EFF SSL Observatory [19] projects collect SSL certificates and publish statistical information about them. The ICSI Certificate Notary also provides a public DNS interface to query its database. These projects collect the certificates by actively probing the websites. As another approach, Huang et al. [29] have used client-side applets implemented in the Facebook website in order to analyze the certificates observed by the client. They have analyzed more than 3 million SSL connections and shared the properties of the observed certificates.

Notary-based solutions are generally criticized for certificate update issues and ineffectiveness in the case when adversaries are close to the server [9].

1.6.3 Pinning-Based Solutions

Pinning methods try to detect certificate substitutions at the client side [31]. Pinning is the process of associating a host with a certificate (or a public key). HPKP creates pins by the user's browsing history [52]. TACK uses server-pushed pins with the TOFU method [55]. Google deploys preloaded pins for various domain names in Chrome [32].

These methods are successful at detecting certificate changes which are possible MITM attacks. They however have some issues about revocation and certificate updates.

1.6.4 Logging-Based Solutions

Sovereign Keys method [18] is a combination of server pinning and logging based methods. Server specifies a public key and logs it at a publicly available append-only log. Losing the private key may end up in losing the domain.

Another example is Certificate Transparency method [37] proposed by Google. Every issued certificate is logged at a publicly available append-only and read-only log with a signed certificate timestamp (SCT). Thus certificates are transparent and verifiable. It is claimed that a MITM attack may be launched by redirecting a client to a specific log or by using a rogue CA [53]. Also revocation seems problematic in CT since the logs are append and read only. In fact, Certificate Transparency does not claim to prevent MITM attacks but to detect them as fast as possible.

1.7 Our Contribution

In this work, we focus on the fact that the SSL servers, in the current trust model, are not able to obtain information on how their certificates are observed at different locations on the network. We propose a complementary solution, the server notaries method, which enables servers to get their certificate views across the Internet. In this way servers will be able to check whether their certificates are observed as expected. Thus detecting a certificate substitution will be possible. Moreover a server may locate the origin of the attack by analyzing certificate views from different vantage points. In order to see how our method performs on the Internet, we have conducted simulation experiments and evaluated our proposal at AS-level Internet topology using publicly available BGP data. We can summarize our primary contributions as follows:

1. We propose the server notaries method, a practical and efficient mechanism that enables servers to observe their certificates from different points on the Internet. Our proposal makes detecting and locating a certificate substitution attack possible.
2. We present a qualitative assessment of advantages and disadvantages of the server notaries method.
3. We present results of simulation experiments conducted using real-life AS-level Internet topology data and evaluate how effective server notaries method can be at detecting a certificate substitution.
4. We propose an effective algorithm to locate the origin of the attack, and evaluate it as a part of our simulation experiments.

The rest of the thesis is organized as follows: Chapter 2 presents the details of our proposal, the server notaries method. Chapter 3 presents our method for collecting and processing the BGP data. Chapter 4 presents network simulation experiments based on real-world BGP data detailed in Chapter 3. Finally, Chapter 5 concludes by summarizing the important results and observations.

CHAPTER 2

OUR PROPOSAL: SERVER NOTARIES

The idea of observing the server certificate from different network vantage points has been used in several proposals to improve the Web PKI trust model. This idea was introduced in Perspectives [61], where Wendlandt et al. defined *notaries* as publicly available semi-trusted hosts deployed at various locations on the network. The main idea is that after a client obtains the server certificate in the usual way, it may compare the received certificate with the server certificate obtained from a notary's network point of view. A difference between the certificates may indicate a certificate substitution. Different variants of notaries have been used in several different protocols. Similar proposals such as Convergence [43], DoubleCheck [2], and CrossBear [28] followed a similar method to enhance the Web PKI trust model.

In this chapter, we introduce a complementary way of using notaries for detecting fake certificates and MITM attacks over the network. In our method, notaries are used by SSL servers rather than clients, hence the name is *server notaries*.

2.1 Threat Model

Our scenario consists of an SSL server, a number of notaries and an adversary. The server in the scenario may be any kind of generic or special purpose server. It announces a certificate publicly to any client wishing to establish a secure channel. Notaries are pre-deployed publicly accessible semi-trusted hosts located at various network points and they are managed by different entities. We assume that the server has already obtained the current list of active notaries and their public keys, as we will explain later.

Our threat model considers an adversary who is able to modify the network traffic flowing over itself. Aim of the adversary is to eavesdrop and tamper with this traffic by executing non-selective MITM attacks against the server. In order to perform such an attack, the adversary may use one of the following methods:

- Obtaining a forged certificate for the server's domain name that is signed by a trusted CA or sub-CA.

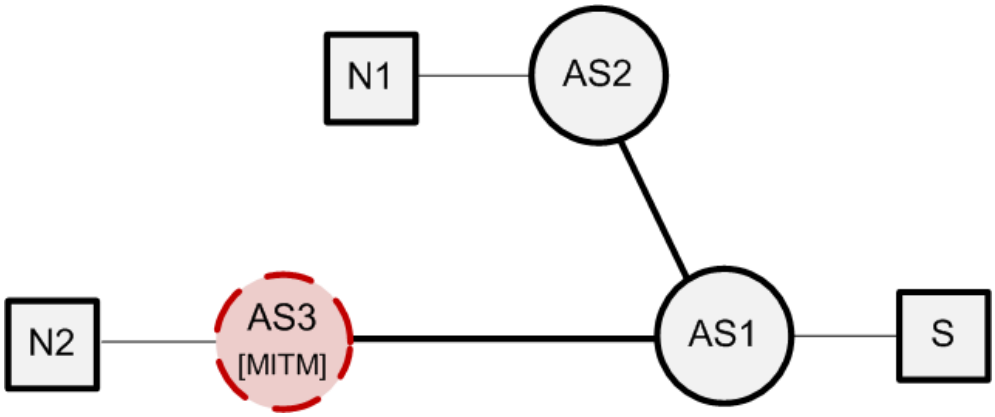

Figure 2.1: A local MITM attack scenario showing AS level network paths between S (*server*), $N1$ (*notary*) , $N2$ (*notary*). Adversary is located at $AS3$. $N1$ observes the genuine certificate, $N2$ observes a fake certificate. Thus S infers that there exists a misbehaving node between S and $N2$.

- Using a revoked certificate before CRL update occurs and by interrupting OCSP queries.
- Launching an HTTPS downgrade attack.
- Using a certificate, untrusted by root stores (e.g. self-signed).

If the MITM attack is local, i.e. the adversary is located in the vicinity of the client, probably the adversary and the client are at the same subnetwork, the same ISP, or the same country. The adversary may be a governmental entity or the ISP itself. In this scenario, the server observes a fake certificate from the notaries deployed within the attack region and a genuine certificate from the remaining notaries. This scenario makes locating the adversary possible. Such an attack scenario is represented in Figure 2.1.

If the adversary is located at a network point close to the server, almost all network paths between the server and the notaries include the adversary. Hence the server will mostly observe a fake certificate from the notaries. The server should check its local network or inform its ISP about the issue.

Our threat model does not consider attacks exploiting implementation or configuration errors. Also we assume that the server is not compromised and is a trusted participant. The notaries are semi-trusted participants. We assume that the adversary is not able to break cryptographic primitives; i.e. the adversary cannot tamper with the data that provides authentication, encryption, or integrity.

2.2 Protocol Details

Server notaries method is based on the exchange of *observation request-response* messages between the server and the notary. The message transaction is given below and demonstrated in Figure 2.2.

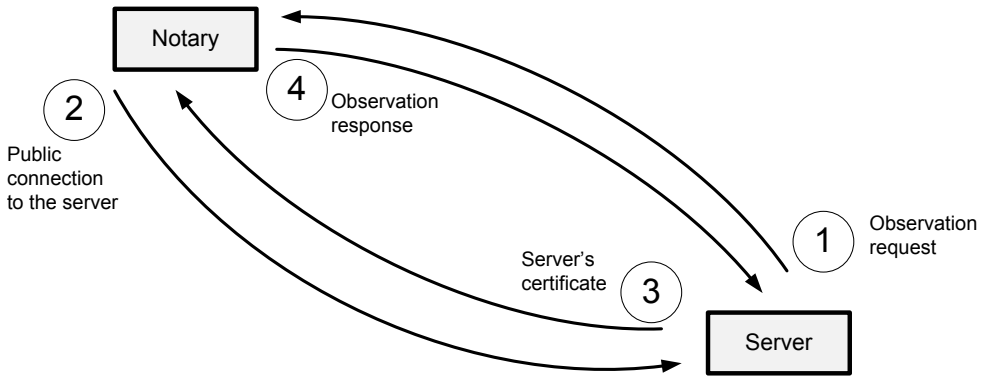

Figure 2.2: Server notaries method overview: (*1*) Server sends an observation request to the notary over secure channel. (*2*) Notary connects to the server over public channel. (*3*) Server sends its certificate. (*4*) Notary sends observation response including the received certificate to the server.

1. Server selects a set of notaries from its notary list and initiates the protocol by sending an *observation request* to these notaries over a secure channel.
2. After receiving the observation request, a notary establishes a connection to the server as any SSL client would do.
3. The notary receives the server's certificate. If there exists an active adversary through the network path between the server and the notary, the notary will receive a fake certificate.
4. Notary sends the signed *observation response* to the server over the previously established secure channel. The observation response includes the observed certificate.

Server notaries method enables servers to *detect* the certificate substitution and *locate* the origin of the attack. If the server receives an unexpected certificate, this is a sign of a certificate substitution between the server and the notary. Hence the server is able to detect a possible MITM attempt or a misissued certificate. Moreover the server is able to locate the network point where the certificate substitution occurs. Spotting the possibly misbehaving nodes through the network may be achieved by comparing the network paths between the server and multiple notaries.

Our proposal does not increase the complexity of the current system. Servers are expected to make periodical probes through the notaries. This can be implemented by minor changes on the server side. Clients are not a part of this method and will remain unmodified.

Similar to other notary-based solutions [61, 28], the server side implementation will include the contact information of a bootstrapping node which will be used to obtain

an active list of notaries and their public keys so that the communication between the server and the notaries are secured.

As a final remark, we would like to note that although we have focused on detecting MITM attacks targeting the Web PKI, server notaries can be used in order to track the view of any certificate or public key served by other processes, such as SSH, as well.

2.3 Discussion

The current Web PKI model is heavily used by billions of users everyday. It is not possible to interrupt the model and to change it by setting a "Flag Day". Hence a viable solution should propose a smooth, gradual transition. It would better include a transition period that interoperates with the current model at least for a while. Server notaries method proposes a quick fix for the vulnerabilities observed in the Web PKI trust model; our proposal would aid servers to mitigate certificate substitution attacks until a final consensus is reached.

The number of participating entities on the Internet is increasing every day. A potential solution should scale as the Internet grows and any participant should be able to use it. For instance, embedding public keys into browsers (preloaded pins) aided researchers in detecting several incidents [33, 35, 36]. However it is not feasible to embed each and every SSL public key in the world into the browsers. On the other hand, the solution should not require every one in the world to participate in order to work properly. For instance, Certificate Transparency enables detecting forged certificates for the participating CAs. It is not applicable, however, to non-participating CAs. Similarly, DANE requires DNSSEC to be deployed at every DNS server worldwide. Thus it can be stated that these solutions are limited by the degree of deployment. It is not the case for server notaries method as any server is able to use it and observe its certificate throughout the Internet. Also it does not require every entity to participate.

Complexity is the enemy of security. The more components a solution has, the harder it is to make it secure. The solution should propose a practical method which does not introduce complex components. Also, it should require as few changes as possible at the server and client sides. Servers, using the server notaries method, will make periodical probes to the notaries. This can be implemented by minor changes on the server side. Notaries can be deployed worldwide using cloud infrastructures. Clients will remain unmodified.

Another issue at the client side is the privacy. In the current model, whenever a client visits a website over SSL, the client's browser queries the CA's OCSP responders to verify that the server certificate is not revoked. Hence, the browsers already leak information about the client's SSL browsing history. Similarly some notary-based solutions suffer from privacy issues. The proposed solution should not introduce additional privacy issues. As clients are not a part of the server notaries method; it does not introduce any privacy issues.

Some of the notary-based solutions solve the privacy issues by anonymizing the com-

Table 2.1: Evaluation of server notaries method with respect to the criteria Clark and Van Oorschot used in [9]. The same structure is kept where the properties are listed as *columns*, • denotes a fulfilled property, and ○ denotes a partially fulfilled property.

Primitive	Security Properties Offered							Evaluation of Impact on HTTPS									
	A				B		C	Security & Privacy				Deployability				Usability		
	Detects MITM	Detects Local MITM	Protects Client Credential	Updatable Pins	Detects TLS Stripping	Affirms POST-to-HTTPS	Responsive Revocation	Intermediate CAs Visible	No New Trusted Entity	No New Traceability	Reduces Traceability	No New Auth'n Tokens	No Server-Side Changes	Deployable without DNSSEC	No Extra Communications	Internet Scalable	No False-Rejects	Status Signalled Completely	No New User Decisions
Key Pinning (Client History)	○	○	○					•	•		•	•	•	•	•				
Key Pinning (Server)	○	○	○					•	•				•	•	•	•		•	
Key Pinning (Preloaded)	•	•	•	•				○	•		•	○	•	•		•	○	•	
Key Pinning (DNS)	•	•	•	•				○	•	•		○		•	•	•	○	•	
Multipath Probing		•		•							•	•	•		•		•		
Channel-bound Credentials			○					•	•		•		•	•	•	•	○	•	
Credential-bound Channels			○					•	•		•		•	•	•	•	○	•	
Key Agility/Manifest				•				•	•				•	•	•	•	•	•	
HTTPS-only Pinning (Server)					○	○		•	•				•	•	•	•		•	
HTTPS-only Pinning (Preloaded)				•	•	•		○	•		•	○	•	•		•	○	•	
HTTPS-only Pinning (DNS)				•	•	•		○	•			○		•	•	•	○	•	
Visual Cues for Secure POST						•		•	•		•	•	•		•		•		
Browser-stored CRL							•	○	•		•	•	•	•	•	•	•	•	
Certificate Status Stapling							•	•	•	•			•	•	•	•	○	•	
Short-lived Certificates							•	•	•	•	•		•	•	•	•	•	•	
List of Active Certificate							•	•		•	•	•	•		•	•	•	•	
Server Notaries	•	○		•	•				•		•		•		•	•	•	•	

munication over the Tor network [12, 2], which causes extra latency for every newly observed certificate at the client side. A usable solution should not add extra latency. The server notaries method will just create extra network traffic on the server side which will not constitute a latency problem.

Notary-based solutions and pinning methods may produce false positive warnings for server farms with multiple different certificates or for websites updating their certificates frequently [9]. Users are expected to make a final decision in such cases. There are also MITM attack detection methods proposed to be used by tech savvy users [28]. A solution may give feedback to the user in case of a suspicious case. However it should not fully depend on end user decisions. Our proposal expects a decision from the server. As the server has the genuine certificate, it can make a final decision for the observed certificate easily.

The deployment of the notary nodes across the Internet is a major issue of our protocol. As noted in [61], independent nodes run by volunteers, like Tor relays, would make an excellent notary infrastructure. Bootstrapping servers can also be implemented à la Tor.

In 2013, Clark and Van Oorschot [9] evaluated the Web PKI trust model enhancement proposals with respect to a basic set of criteria. Table 2.1 presents an evaluation of our proposal according to these criteria.

An adversary may obtain a certificate for a domain, accepted by browsers, without the consent or the knowledge of the domain owner. Thus the adversary can launch a

MITM attack by substituting the genuine certificate with his own by modifying (e.g. DNS hijacking, on path interception in the vicinity of the server) the traffic through the server. Methods detecting such attacks fulfills the "Detects MITM" property, denoted by • . If the method includes a risk or requires TOFU, it partially fulfills the property, denoted by ○ . Our proposal fulfills the "Detects MITM" property since a server is able to detect a global certificate substitution attack by utilizing server notaries method.

If the adversary is able to intercept the traffic locally (e.g. manipulating local DNS cache or on path interception in the vicinity of the client) than this MITM attack is called local. Proposals detecting such attacks fulfill the "Detects Local MITM" property. If the method includes a risk or requires TOFU, it partially fulfills the property. Our proposal partially fulfills this property since detection of such an attack requires a notary to be deployed within the vicinity of the client.

Authentication credentials (usernames, passwords etc.) are usually transmitted through an SSL connection. Some proposals focus on protecting these credentials. Such proposals fulfill the "Protects Client Credential" property. Similarly, proposals requiring TOFU partially fulfills this property. There exists no protection against any kind of credential theft in our proposal, so it does not fulfill this property.

Some of the proposals produce false negatives if a server updates its public key, changes its CA, or uses multiple certificates. If a proposal is resilient against such cases, it fulfills the "Updatable Pins" property. Our method fulfills this property as the server is able to compare the observed certificate with the actual one.

Adversaries may launch an SSL stripping attack [47, 41] which simply downgrades an HTTPS connection request to an HTTP one. A proposal detecting such attacks fulfills the "Detects TLS Stripping" property, and partially fulfills the property if it requires TOFU. Our proposal fulfills this property, as it would detect such attacks launched between a notary and the server.

A proposal that prevents HTTPS requests to be submitted over HTTP (e.g. by forcing a policy) fulfills the "Affirms POST-to-HTTPS" property. If a proposal requires TOFU it partially fulfills the property. Our proposal does not fulfill this property as it does not have an enforcement on the client side.

The "Responsive Revocation" property evaluates whether the proposal is successful at detecting revoked certificates when CRLs and OCSP responses are not available. The method is expected to keep the log or the history of certificates to fulfill this property. Hence our proposal does not fulfill this property.

The "Intermediate CAs Visible" property checks whether a proposal enables the user to observe every intermediate CA anytime. This property requires all issued certificates to be logged, hence it is not applicable to our method.

The current Web PKI model deploys CAs as the trusted entities. Some proposals introduce new trusted entities (e.g. notaries) to the infrastructure. If a proposal does not introduce such an entity, it fulfills the "No New Trusted Entity" property. If the responsibility of the current set of trusted entities are expanded, the property is partially

fulfilled. Our method utilizes notaries, thus it does not fulfill this property.

As discussed earlier, browsers already leak information about the user's HTTPS browsing history (e.g. through the usage of OCSP responders). If the proposal does not leak information to any additional parties, it fulfills the "No New Traceability" property. Apart from other notary-based approaches, our proposal fulfills this property since it does not include user interaction.

Similar to the previous property, if the proposal eliminates a set of entities which can keep track of user's HTTPS browsing history, it fulfills the "Reduces Traceability" property. Our proposal does not have an effect on these entities, hence it does not fulfill this property.

Some of the proposals introduce new authentication tokens such as pins or signed OCSP responses. Issuing, updating, revoking these tokens, and providing the integrity of them make the proposal more complex. If a proposal does not introduce such tokens, it fulfills the "No New Authentication Tokens" property. Our proposal fulfills this property as it does not require additional tokens.

Some of the proposals require the SSL server to participate in the solution or furthermore to change the way it implements the SSL communication. If the proposal does not require any kind of server-side change or server participation, it fulfills the "No Server-Side Changes" property. If the proposal requires just the participation of the server and does not require code change on the server side, it partially fulfills this property. Our proposal does not fulfill this property as it requires both the participation of the server and minor code changes on the server.

If a proposal does not depend on DNSSEC, it fulfills the "Deployable without DNSSEC" property. Our proposal does not require DNSSEC to be deployed, so it fulfills this property.

The "No Extra Communications" property checks whether a proposal introduces extra communication which blocks the completion of the usual communication flow (e.g. waiting for the notary response). Our proposal does not block the completion of the usual communication flow between the client and the server. However, since it generates additional traffic through the server, we assume that it does not fulfill this property.

It is expected that a solution should scale well so that all SSL servers and more would be able to participate. Proposals satisfying this condition fulfills the "Internet Scalable" property. Our proposal is a scalable solution, hence it fulfills this property.

Usability properties should also be evaluated for the proposals. If the proposal does not reject a genuine server certificate, it fulfills the "No False Rejects" property. Our proposal, from the server point of view, fulfills "No False Rejects", since the server has the genuine certificate.

In the Web PKI trust model, an HTTPS connection is successful if the certificate is accepted by the browser. Thus the final decision does not depend on the implementation of the proposal. For instance, a TOFU-based proposal does not fulfill the "Status Signalled Completely" property if the trust decision depends on how the proposal is

implemented. It partially fulfills this property, if the server enrollment is required to clarify the basis of trust. Our proposal fulfills this property, since it is applicable to all servers independent from how it is implemented.

Proposals not fulfilling the "No False Rejects" property, gives the final trust decision to the user. If the proposal automates this process and does not require user participation, it fulfills the "No New User Decisions" property. Our proposal fulfills this property since it does not depend on user decisions.

The evaluation in Table 2.1 shows that a unique solution fulfilling all the properties does not exist yet. Unless precautions are taken immediately, it is no surprise to see more security incidents in the near future. Additionally, we won't be replacing the current Web PKI model at one night. Hence, it makes sense to deploy our proposal until a final consensus is reached since it is a certificate monitoring oriented tool and it is easily applicable to the current Web PKI trust model.

Yet another discussion topic is the comparison between the traditional client oriented notary approach and our proposal. Former is criticized for not being effective against adversaries close to the server and for producing false positives in some cases (e.g. certificate updates). Our proposal is effective against the adversaries in the vicinity of the server. False positives are not the case for our proposal since the server has the genuine certificate. As our proposal overcomes these drawbacks, it would be wise to deploy a client oriented notary approach using the same notary infrastructure which would make both approaches more powerful.

CT is one of the most discussed proposals among all other solutions. Currently Google encourages CAs and other entities to participate in the solution. CT is being built on the current model by introducing new components with different roles. This will end up with a more complicated structure. Also, it requires each and every entity to be a part of the new model in order to work properly. On the other hand, our proposal introduces server notaries to be used by servers as a monitoring node for CA issued certificates. It just requires a set of server notaries to work properly. Moreover, server notaries are suitable to be used for adding certificates to the CT log, for CAs that fail to do so. Thus, if CT becomes a commonly accepted solution one day in the future, our proposal will be a useful extension complementing the CT infrastructure as well.

CHAPTER 3

DATA COLLECTION AND ANALYSIS

Recently several network researchers are trying to analyze properties of Internet topology. It is clear that by analyzing the properties of Internet topology, more efficient algorithms may be designed, more realistic network simulations may be generated and the future of the Internet may be better interpreted. However, as being composed of several smaller networks apart, it is hard to define topology evolution of the Internet. Researchers are mainly focused on Internet topology at Autonomous System level.

An Autonomous System (AS) is defined as a connected group of one or more IP prefixes run by one or more network operators which has a single and clearly defined routing policy [27]. Each AS is represented with a globally unique number associated with it which is called the Autonomous System Number (ASN). This number is used for the exchange of routing information with the neighboring ASes and also as an identifier for the AS globally.

Each AS is controlled by a single administrative entity. There exists an exception for the use of private ASNs. For instance, an ISP may use private ASNs in order to connect different organizations using different prefixes to the Internet. However, only the ISP's officially registered ASN is observed from the Internet.

AS numbers were defined by 16-bit integers until 2007 which allowed 65536 assignments at maximum. 32-bit AS numbers were introduced by RFC 4893 [59] in 2007 which was updated by RFC6793 [60] in 2012. As seen in Figure 3.1 the number of unique autonomous systems was around 10000 in 2001. Currently it is more than 50000.

Border Gateway Protocol (BGP) is the current de facto standard for inter-AS routing. BGP is defined as the protocol for exchanging routing and accessibility information between ASes [48]. The network reachability information includes the list of ASes to be traversed through a destination. Thus this information may be used to construct the AS level map of the Internet topology.

A BGP speaker is a network node which announces and receives routes via BGP. It receives routing updates from other peers, processes this information for local use, and advertises selected routes to other peers based on predefined policies. BGP speakers store BGP information as a special type of database called the BGP *Routing Information Base (RIB)* to perform their functions.

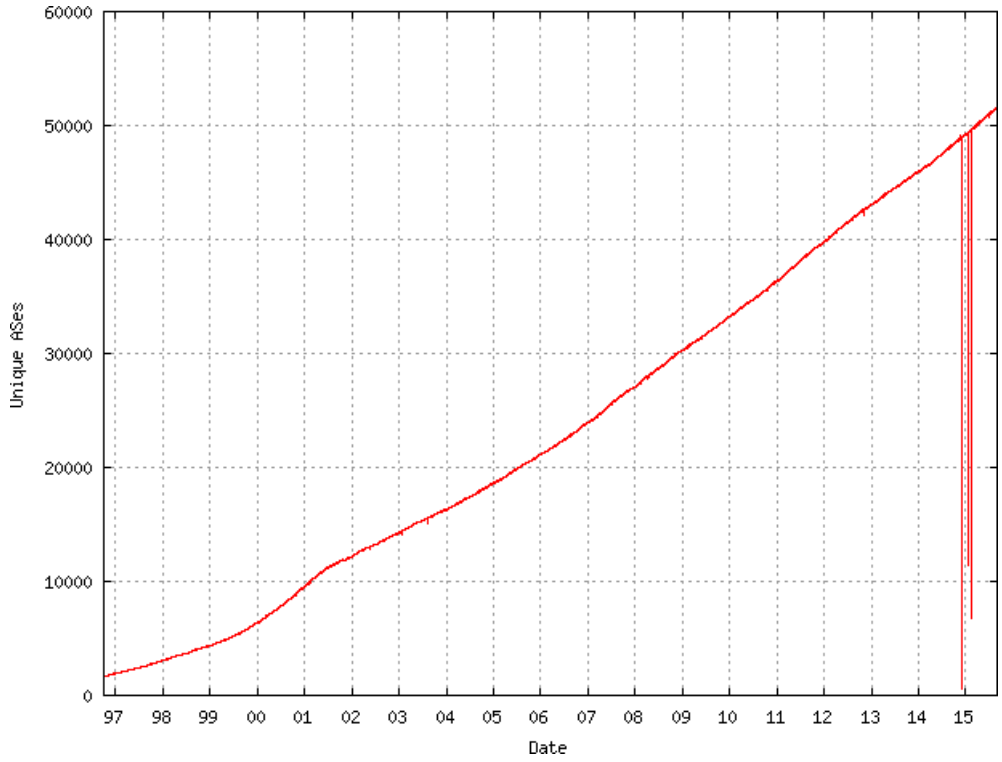

Figure 3.1: Increase in the number of unique ASes [4]

BGP is a destination oriented routing protocol. In other words a BGP speaker checks for the destination address of a packet and selects the next hop of the packet according to this address from its routing table. Unfortunately this behavior makes some policies, such as source based routing, inapplicable using BGP.

In this chapter, we focus on the details of data collection and analysis procedures of our simulation experiments. We present how we have processed the publicly available BGP data. Additionally, we share the current AS-level statistics of the Internet.

3.1 Processing the BGP Data

Throughout this work we used the BGP data provided by the University of Oregon Route Views Project [57]. This project aims publishing data about the global view of the Internet using routing information. This project gives real time access to the routing data publicly.

Routeviews data have been used in several projects. An already completed one is the NLANR [46] project which had used the data for AS path visualization and IPv4 address space utilization. In a more recent study, CAIDA [8] has been using Routeviews data to generate geographical location of hosts in conjunction with the NetGeo [45] database. CADIA AS Relationships [7] project is another example. This project investigates business agreements between ASes based on customer/provider/peer relations.

```
#!/usr/bin/env perl
use strict;
use warnings;
use Data::Dumper;

# Net::MRT - Perl extension for decoding
# RFC6396 Multi-Threaded Routing Toolkit (MRT)
# Routing Information Export Format
# http://search.cpan.org/dist/ \\
#          Net-MRT-0.0303/lib/Net/MRT.pm
use Net::MRT;

open(C, '<', 'rib.20150809.0800');
binmode(C);
while (my $decode = Net::MRT::mrt_read_next(C)) {
  print Dumper(\$decode)."\n";
}
```

Figure 3.2: Perl code snippet for decoding binary MRT files

There are collectors deployed worldwide which gather the routing data. They have established BGP connections with several BGP peers. Note that these collectors are working in passive mode, i.e. they collect routing information via peerings however they do not announce any prefix to the Internet. Collectors' main purpose is to observe advertised AS paths through the Internet. Although it is not feasible to deploy a collector at every AS for observation, it is shown that the public BGP information is enough to capture relatively complete AS level Internet topology [20].

By August 2015, there are 437 peering to 188 distinct ASes using 19 collectors in total [56]. It is observed that some of the collectors are deployed within Tier-1 networks.

Collectors dump raw BGP routing information since 1997. There have been format changes up to now. Currently the data may be obtained for two hours periods as compressed (.bzip2) files which include binary RIB files.

The binary RIB files may be decoded using several libraries. We used Perl Net::MRT extension [3] as given in Figure 3.2. The decoded output is in the Multi-threaded Routing Toolkit (MRT) standard as defined in RFC6396 [6]. MRT record format is developed to encapsulate, export and archive routing information such as routing protocol messages, state changes and routing information base contents in a standardized manner. Hence researchers and engineers are able to study the network behavior by analyzing routing protocol transactions and routing information base snapshots.

In order to specify the type of the MRT formatted file, the *type* field should be checked. In our case MRT formatted files are in *TABLE_DUMP_V2* type (type=13), which is

the updated version of *TABLE_DUMP* type, supporting 4-octet ASNs and increasing the efficiency by employing the peer index table. Our MRT files include records for *subtype*s *PEER_INDEX_TABLE* (subtype=1), *RIB_IPV4_UNICAST* (subtype=2), and RIB_IPV6_UNICAST (subtype=4). We are interested in observing unicast IPv4 traffic so we extracted the data *RIB_IPV4_UNICAST* subtype from the whole data set. A snapshot of the MRT files is presented in Appendix A.

We have downloaded and parsed the data set (MRT-formatted full-table RIBs Routing Information Base, i.e., BGP dumps.) for 9 August 2015 (08:00) for the vantage points: Oregon IX, Equinix Ashburn, ISC/PAIX, KIXP, LINX, DIXIE/WIDE, RouteViews-4, Sydney, and São Paulo. The data includes BGP tables collected from 188 distinct ASes world wide. The raw data includes misleading information such as repetition of AS paths or loops inside AS paths. We have discarded data sets that are truncated or having limited IP space. We have removed invalid paths like loops or repetitive ASes and duplicate paths. After these steps we have obtained the *AS path dataset* including more than 11 million AS paths from 124 distinct ASes destined to almost all ASes observed worldwide.

3.2 Internet Topology Today

Business relationships between ASes have economical and technical importance. CAIDA AS Relationships project [7] processes publicly available BGP information and presents the AS relationship dataset. This process is based on the *customer cone* approach [15, 16]. The customer cone of an AS is defined as the set of ASes that can be reached from each AS following only its customer links.

In general AS relationships are classified into three main categories namely provider to customer (p2c), peer to peer (p2p) and sibling to sibling (s2s). According to CAIDA, this relationship is associated with the flow of cash [7]. A customer pays its provider in order to have connection to the Internet. In other words provider allows transit of customer's data over itself. However customer does not allow transit traffic of its providers over itself. Peer-to-peer connection is defined as the exchange of traffic between the respective customers of each peer free of charge. This kind of connection may be observed between small ISPs who cannot afford additional Internet services for better connection or between administrative domains who wish to deploy a backup connectivity. As the last item a s2s link is the connection between two nodes who are administratively belonging to the same ISP.

A sample AS level topology is represented in Figure 3.3. In the sample topology, ISP D and E are customers of ISP B. ISP F is the customer of ISP C. ISP B and C are providers for the ISPs below them and customers of ISP A. Additionally they are peers of each other. ISP A is the provider of ISP B and C.

CAIDA AS relationship project publishes relationship dataset once a month. A sample from the August 2015 dataset is presented in Figure 3.4. Each line represents a relationship between two ASes where first two numbers are the AS numbers and the final number (0:peer or -1:p2c) states the relationship between these ASes. If it is 0 the

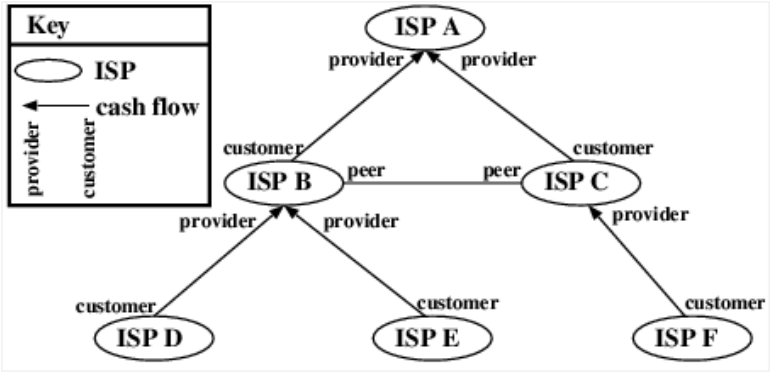

Figure 3.3: AS relationship sample topology [7].

```
...
24|11537|0
26|47065|-1
32|3561|0
32|3671|-1
32|46750|-1
...
```

Figure 3.4: CAIDA AS-relationship dataset sample

ASes are peers of each other, otherwise if it is -1 then the first AS is the provider of the second. For the given sample, AS 24 and AS 11537 are peers of each other and AS 26 is the provider of AS 47065.

AS degree distribution using AS relationship dataset is given in Figure 3.5. Degree one ASes are the leaf nodes. It is remarkable that there are more degree two ASes than degree one ASes. This is almost the same result as the ones found in [26] by Govindan et al. and in [40] by Magoni et al.

Table 3.1 presents statistical information gathered parsing both the AS relation dataset and the BGP data. By August 2015 there are 51260 ASes observed. Average provider, customer, and peer numbers per node are 1.96, 1.92 and 3.73 respectively. Using BGP dataset we have observed 625988 /24 prefixes announced in total. This implies 12.1 /24 prefix per an AS.

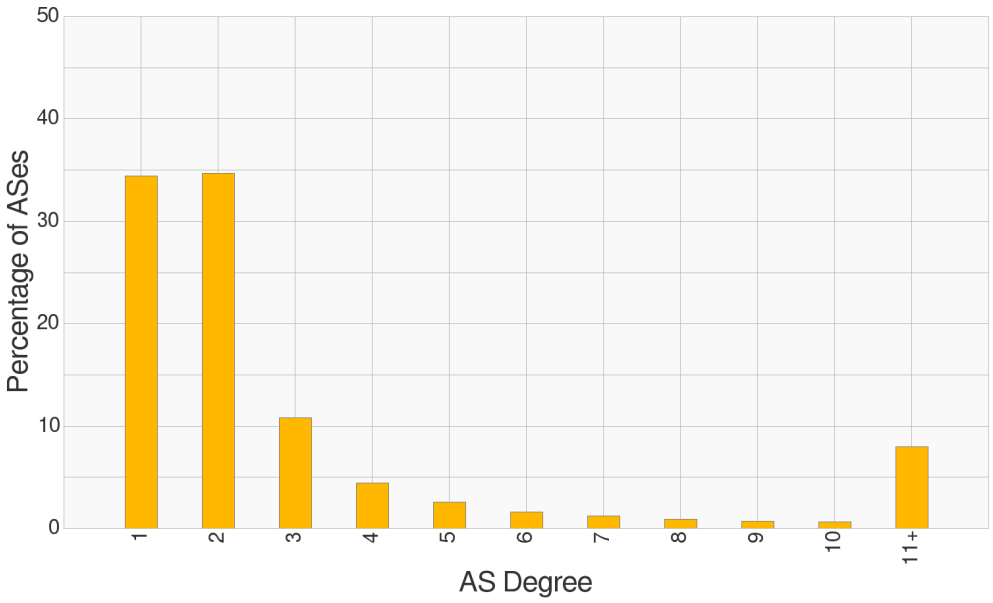

Figure 3.5: Percentage of ASes (*y-axis*) with respect to the degree of ASes (*x-axis*).

Table 3.1: Statistical details on AS-level Internet topology

Average provider per AS	1.96
Average customer per AS	1.92
Average peer per AS	3.73
Average degree	7.66
Prefixes announced	625988
Average prefix per AS	12.1

CHAPTER 4

SERVER NOTARIES SIMULATIONS

4.1 Detection

Server notaries method has two types of components namely the *servers* and the *notaries*. We consider the AS-level Internet topology where BGP policies determine the AS paths available between two ASes.

As for the *servers*, we used the *collectors* of the AS path dataset described in Chapter 3. Recall that we have obtained AS paths sourcing from 124 distinct ASes to almost all ASes observed in the Internet. Hence, we have decided to use the 124 distinct source ASes as our servers in the simulation.

An important question regarding the deployment of the server notaries method is how to distribute the notaries over the Internet for an effective utilization. An intuitive idea for deployment is to put the notaries at the highly-connected ASes. To choose the notary ASes, we sorted all ASes in descending order with respect to the following five AS features and took a given number of highest ranking ones. Last three items are related to the business agreements between ASes which are typically confidential but may be inferred from BGP data [39, 22].

- **Degree**: The number of ASes directly connected to an AS.
- **Prefix**: The number of prefixes an AS announces.
- **Provider:** The number of providers an AS has.
- **Customer:** The number of customers an AS has.
- **Peer:** The number of peers an AS has.

We used RouteViews BGP data to calculate the number of announced prefixes per AS. We used CAIDA AS Relationship dataset [7], which presents the AS relations as provider-to-customer or peer-to-peer, to calculate the remaining AS features.

We say that ASes observed between the server AS and the notary AS are *covered* by the notary for the server. Covered ASes are critical at detecting adversaries. Assume

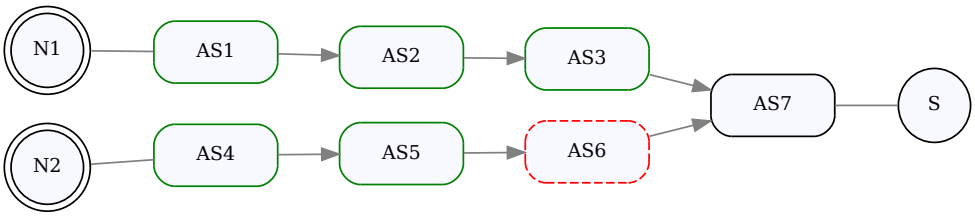

Figure 4.1: Sample set of AS paths including the server (*S*) and the notaries (*N1*, *N2*). An adversary is located at *AS6*. *N1* observes the genuine certificate. *N2* is effected by the adversary on its path to *S* and observes the fake certificate.

an adversary is located at one of the covered ASes and substitutes the server certificate by a forged one. Then the server would detect the adversary by querying the respective notary's view since the notary observes the forged certificate.

A sample scenario is presented in Figure 4.1. The server *S* is located at AS7 and the notaries *N1* and *N2* are located at AS1, AS4 respectively. AS1, AS2, AS3, and AS7 are covered by *N1*. AS4, AS5, AS6, and AS7 are covered by *N2*. Server detects the adversary, located at AS6, by querying *N2*.

4.1.1 Performance Metrics

We define the following performance metrics over the AS path dataset. Hereafter s denotes an SSL web server AS, n_i denotes a notary AS, and N denotes the set of all notary ASes.

$\boldsymbol{CAS(s, N)}$ **:** "Covered AS" (CAS) is the number of distinct ASes observed through the AS paths between s and all notaries in N.

$\boldsymbol{TAS}$ **:** "Total AS" (TAS) is the number of distinct ASes observed in the AS path dataset.

In order to calculate $CAS(s, N)$ value for one server s, we scanned the AS path dataset for paths having s and n_i as the first and last ASes, $\forall n_i \in N$. We counted the number of distinct ASes observed on these paths and found the $CAS(s, N)$ value. After calculating the $CAS(s, N)$ values for all servers, we calculated their mean value CAS. Using CAS and TAS values, we calculated $CAS\ Ratio$ as follows:

$$CAS\ Ratio = \frac{CAS}{TAS} \tag{4.1}$$

This value gives the ratio of covered distinct ASes using the set of notary ASes N.

$\boldsymbol{CASH(s, N)}$ **:** "Covered AS Hit" (CASH) is the total number of occurrences (including multiple counts) of covered ASes in the AS path dataset.

$\boldsymbol{TASH}$ **:** "Total AS Hit" (TASH) is the total number of occurrences (including mul-

tiple counts) of all ASes in the AS path dataset.

We found covered ASes by n_i for s, $\forall n_i \in N$. Then we counted the occurrences of these ASes in the AS path dataset and found $CASH(s, N)$ value. After calculating $CASH(s, N)$ values for all servers, we calculated their mean value $CASH$. Using $CASH$ and $TASH$ values, we calculated $CASH\ Ratio$ as follows:

$$CASH\ Ratio = \frac{CASH}{TASH} \tag{4.2}$$

$CASH\ Ratio$ value represents how frequent the covered ASes are observed over the AS path dataset. This is also the probability that a random AS path includes a covered AS. If an adversary, launching a MITM attack by certificate substitution, is located at one of the covered ASes, it will be detected using our method. Hence, we interpret $CASH\ Ratio$ as the *probability of detecting an adversary* at AS-level.

4.1.2 Results

The contribution of this simulation is twofold. Firstly, we evaluate how successful server notaries method is at detecting certificate substitution attacks. Secondly, we analyze the effect of several AS features on AS selection for notary deployment.

$CAS\ Ratio$ values are given in Figure 4.2. This figure shows that top n ASes with the highest number of providers will cover a larger portion of the network than other alternatives, for a given number n. For instance, top 200 ASes from the "provider" list cover approximately 1.5% of all ASes where top 200 ASes from the other lists cover less than 1% of all ASes.

$CASH\ Ratio$ values, which measure the probability of detecting an adversary, are presented in Figure 4.3. The results are very promising. By deploying notaries at top 200 ASes from the "degree" list, probability of detecting an adversary at the AS level is more than 50%. The simulation results show that it is better to deploy notaries at ASes with higher degrees in order to have a higher probability of detecting adversaries. By deploying notaries at the top 2000 ASes from the degree list, the $CASH\ Ratio$ becomes 70%.

4.2 Locating the Origin of the Attack

Until now we have discussed how a server detects a certificate substitution attack. Our aim is to make our method more powerful by enabling servers to locate the origin of the attack. Server notaries method, as stated previously, is based on the exchange of observation request-response messages. Notaries send observation response messages to the server. These messages include the observed server certificate and the AS paths from the notaries to the server. The AS path data is the key that makes locating the origin of the attack possible.

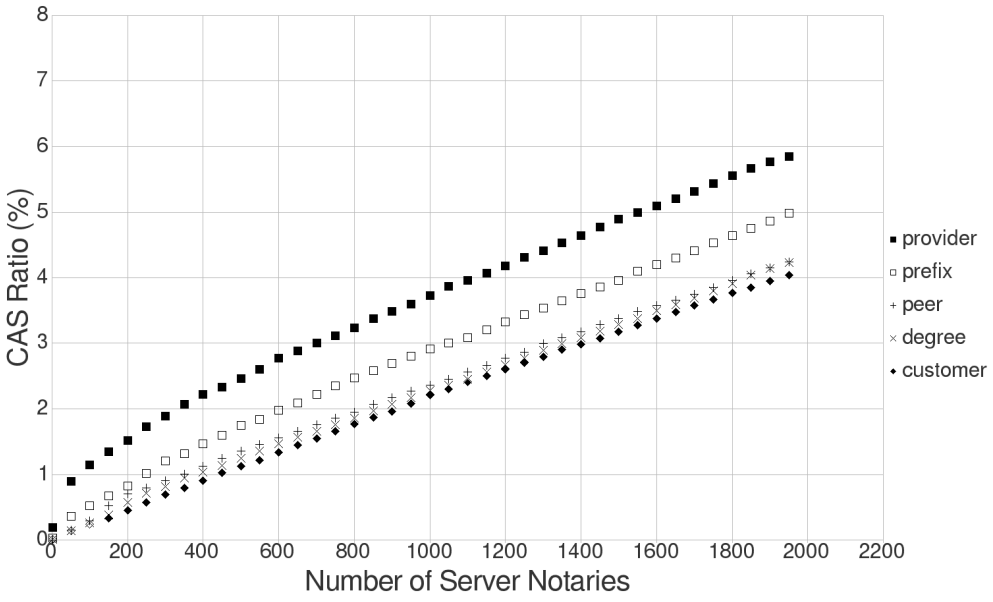

Figure 4.2: Percentage of CAS Ratio (*y-axis*) with respect to the number of notaries (*x-axis*), selected according to the AS features given in the legend.

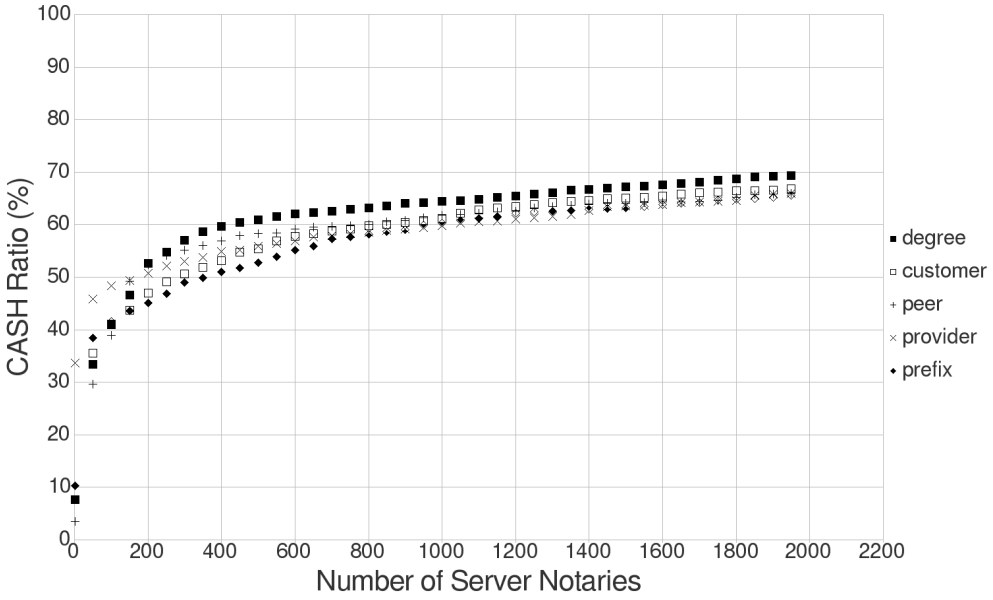

Figure 4.3: Percentage of CASH Ratio (*y-axis*) with respect to the number of notaries (*x-axis*), selected according to the AS features given in the legend.

There may be different approaches to locate the adversary using the observation responses. The first idea is to separate the AS paths received with respect to the observation responses as genuine and fake AS sets. This approach guarantees that the adversary is a member of the fake AS set. However this set would have too many members which makes spotting the adversary harder.

An improvement to this rough idea is to search for two AS paths which have ASes in common and where the genuine certificate is observed from one of them, fake certificate from the other one. This case reveals the fact that there is something wrong with the ASes remaining in the difference set of these two AS paths. This idea requires the notaries to be located densely throughout the network. A similar approach has been used in the Crossbear proposal [28].

Yet another improvement of this idea can be obtained by assigning scores to ASes according to the observation responses received. Then the AS with the lowest (or highest) score will be taken as the most likely origin of the attack. Below we describe an algorithm based on this approach.

4.2.1 Origin Location Algorithm

Once the server detects a certificate substitution attack, it makes a list of ASes observed within the AS paths received from the notaries. The server keeps a *rating value*, which is initially set to 0, for each AS observed. Then the server analyzes each AS path and update the rating value of ASes on the path with respect to the observation response for that path: If a fake certificate is observed through the path, the server decreases the rating value of ASes on the path by 1. Otherwise, the server increases the rating value of ASes on that path by 1. After completing the update phase, the server sorts the ASes with respect to their rating values in descending order, and returns the set of ASes having the smallest rating value. The pseudo code for this algorithm is given in Algorithm 1.

The algorithm is based on two basic facts: (1) An important aspect of BGP is that the AS path itself is an anti-loop mechanism. Routers will not import any routes that contain themselves in the AS path [48]. Hence a path cannot include an AS more than once. (2) The adversary defined in the threat model launches a non-selective certificate substitution attack.

(1) and (2) imply that the adversary AS would have the smallest rating value. Recall that the algorithm returns the set of ASes having this value. Hence it is obvious that this set will include the origin of the attack.

The cardinality of the returned set shows how precise the locating process is. Having cardinality one means pinpointing the exact origin of the attack. Cardinality will be greater than one if there exists an AS observed exactly in the same paths as the adversary. It will also have the smallest rating value which will result in false positives for the probable location of the adversary. Most probably this is the case for one of the neighbors (e.g. provider) of the adversary, and it will occur if there are too few paths

Algorithm 1 Locating the Origin of the Attack

```
1: procedure LOCATE–ADVERSARY
2: initialize:
3:     for each as ∈ AS do
4:         rating[as] ← 0
5:     end for
6: update:
7:     for each observation response received do
8:         if the certificate observed is TRUE then
9:             for each as in observation path do
10:                rating[as] ← rating[as] + 1
11:            end for
12:        end if
13:        if the certificate observed is FALSE then
14:            for each as in observation path do
15:                rating[as] ← rating[as] − 1
16:            end for
17:        end if
18:    end for
19: return {as : rating[as] ≤ rating[i] ∀i ∈ AS}
20: end procedure
```

through which a fake certificate is observed.

4.2.2 Methodology

In this simulation, our aim is to measure how effective our algorithm is at locating the origin of a certificate substitution attack. We used the AS path dataset. Recall that this dataset is loop-free and does not include invalid or duplicate paths. For this simulation, we used the collectors of the AS path dataset as the servers. We used top degree ASes as the notaries since we obtained better CASH values using them.

We started by extracting AS paths originating from one server and destined for the notaries from the AS path dataset, i.e. the covered AS paths. As the adversary we selected one of the ASes within these paths. We applied our algorithm to these paths which updated the AS rating values and returned the set of ASes having the smallest rating value. Remember that the cardinality of this set gives how many candidate ASes exist as the most probable origins of the attack.

Then the procedure above is repeated with each AS selected as the adversary, and each collector selected as the server. Finally we calculated the average cardinality values for all these cases.

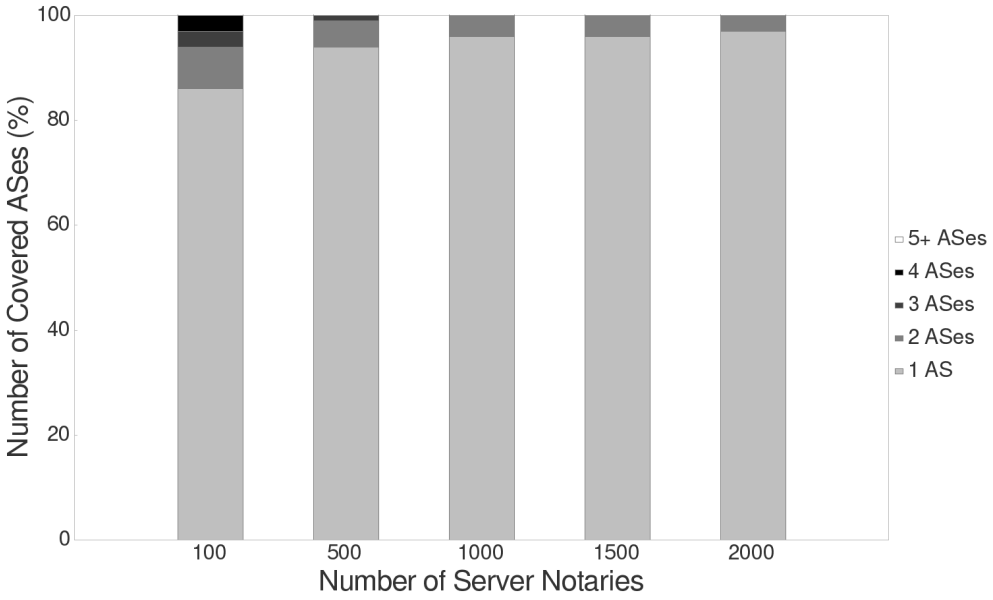

Figure 4.4: Distribution of cardinality of the AS set with the smallest rating value presented using the percentage of covered ASes (*y-axis*) with respect to the number of notaries (*x-axis*).

4.2.3 Results

Distribution of the cardinality of the set of lowest-rated ASes (i.e. the most likely attackers) with respect to the number of notaries is given in Figure 4.4. Recall that having cardinality 1 means the algorithm locates the origin of the attack exactly. This is the case for 86% of the covered ASes when 100 notaries are deployed. Using the same set of notaries, the cardinality is 2, 3, and 4 for 8%, 3%, and 3% of the covered ASes respectively.

It is observed that our algorithm locates the adversary more accurately as the number of notaries increases. When there are 2000 notaries deployed, the adversary would be located exactly in 98% of the attacks, and for the remaining cases there will be two candidate ASes as the probable origin of the attack.

CHAPTER 5

CONCLUSION

Recent incidents have demonstrated the vulnerabilities in the Web PKI trust model. As most of these vulnerabilities remain unsolved, the number of incidents are expected to increase over time.

Current solutions use different approaches to mitigate known vulnerabilities. There exist solutions utilizing the DNS hierarchy, notaries, client browsers, or other infrastructures including log servers. It is possible to say that there will not be a final, elegant solution in the near future by looking at the complexity and deployability issues of the proposed solutions.

Keeping this in mind we have come up with a complementary approach to the current Web PKI trust model. Server notaries is a practical mechanism which enables servers to observe their own certificates using public notaries. Apart from other solutions, server notaries method assigns SSL servers a more active role. This would bring the server administrators into the game as they will try to detect attacks against their servers.

We collected and analyzed real-world AS-level data to build a snapshot of the actual Internet topology. Then we conducted simulation experiments to evaluate detection and locating effectiveness of server notaries method.

The results show that the server notaries method can be effective at detecting a certificate substitution attack and locating its origin. We observed that ASes having higher degrees are better candidates for notary placement. It is more probable to locate the origin of the attack by deploying the notaries at these ASes.

Our work suggests a variety of future research directions. One such direction is to implement the server notaries method and observe how effective it is in the wild. Another one is to conduct simulation experiments of other proposals using the same AS-level topology dataset.

REFERENCES

[1] ISO/IEC 27000:2014(E) information technology — security techniques — information security management systems — overview and vocabulary, 2014.

[2] M. Alicherry and A. D. Keromytis, Doublecheck: Multi-path verification against man-in-the-middle attacks, in *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pp. 557–563, IEEE, 2009.

[3] M. Basunov, Net::MRT - Perl extension for decoding RFC6396 multi-threaded routing toolkit (MRT) routing information export format, 2013, `http://search.cpan.org/dist/Net-MRT-0.0303/lib/Net/MRT.pm`.

[4] T. Bates, P. Smith, and G. Huston, The CIDR Report, August 2015, Available at `http://www.cidr-report.org/as2.0/`.

[5] S. Birkner, The sequence diagram of a man-in-the-middle attack of the diffie-hellmann key agreement., 2006, `https://commons.wikimedia.org/wiki/File%3AMan-in-the-middle_attack_of_Diffie-Hellman_key_agreement.svg`.

[6] L. Blunk, M. Karir, and C. Labovitz, Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format, RFC 6396 (Proposed Standard), October 2011.

[7] CAIDA, AS Relationships, 2015, `http://www.caida.org/data/as-relationships/`.

[8] CAIDA, Center for applied Internet data analysis, 2015, `http://www.caida.org`.

[9] J. Clark and P. C. van Oorschot, SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements, in *Security and Privacy (SP), 2013 IEEE Symposium on*, pp. 511–525, IEEE, 2013, ISSN 1081-6011.

[10] Comodo, Comodo SSL affiliate the recent RA compromise, March 2011, `https://blog.comodo.com/other/the-recent-ra-compromise/`.

[11] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 5280 (Proposed Standard), May 2008, updated by RFC 6818.

[12] DetecTor, `http://www.detector.io`.

[13] T. Dierks and E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, RFC 5246 (Proposed Standard), August 2008, updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627.

[14] W. Diffie and M. E. Hellman, New directions in cryptography, Information Theory, IEEE Transactions on, 22(6), pp. 644–654, 1976.

[15] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley, AS Relationships: Inference and Validation, ACM SIGCOMM Computer Communication Review (CCR), 37(1), pp. 29–40, Jan 2007.

[16] X. Dimitropoulos, D. Krioukov, B. Huffaker, k. claffy, and G. Riley, Inferring AS Relationships: Dead End or Lively Beginning?, in *4th Workshop on Efficient and Experimental Algorithms (WEA)*, pp. 113–125, Springer Lecture Notes in Computer Science, Santorini, Greece, May 2005.

[17] P. Eckersley and J. Burns, The (decentralized) SSL observatory, in *Invited talk at 20th USENIX Security Symposium*, 2011.

[18] EFF, The sovereign keys project, `https://www.eff.org/sovereign-keys`.

[19] EFF, The EFF SSL observatory, 2015, `https://www.eff.org/observatory`.

[20] M. Faloutsos, P. Faloutsos, and C. Faloutsos, On power-law relationships of the Internet topology, SIGCOMM Comput. Commun. Rev., 29(4), pp. 251–262, August 1999, ISSN 0146-4833.

[21] A. Freier, P. Karlton, and P. Kocher, The Secure Sockets Layer (SSL) Protocol Version 3.0, RFC 6101 (Historic), August 2011.

[22] L. Gao, On inferring autonomous system relationships in the Internet, IEEE/ACM Trans. Netw., 9(6), pp. 733–745, December 2001, ISSN 1063-6692.

[23] Google, Improved digital certificate security, September 2015, `https://googleonlinesecurity.blogspot.com.tr/2015/09/improved-digital-certificate-security.html`.

[24] Google, Proactive measures in digital certificate security, December 2015, `https://googleonlinesecurity.blogspot.com.tr/2015/12/proactive-measures-in-digital.html`.

[25] Google, Sustaining digital certificate security, October 2015, `https://googleonlinesecurity.blogspot.com.tr/2015/10/sustaining-digital-certificate-security.html`.

[26] R. Govindan and A. Reddy, An analysis of internet inter-domain topology and route stability, in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, volume 2, pp. 850–857 vol.2, Apr 1997, ISSN 0743-166X.

[27] J. Hawkinson and T. Bates, Guidelines for creation, selection, and registration of an Autonomous System (AS), RFC 1930 (Best Current Practice), March 1996, updated by RFCs 6996, 7300.

[28] R. Holz, T. Riedmaier, N. Kammenhuber, and G. Carle, X. 509 forensics: Detecting and localising the SSL/TLS men-in-the-middle, in *Computer Security–ESORICS 2012*, pp. 217–234, Springer, 2012.

[29] L. S. Huang, A. Rice, E. Ellingsen, and C. Jackson, Analyzing forged SSL certificates in the wild, in *Security and Privacy (SP), 2014 IEEE Symposium on*, pp. 83–97, IEEE, 2014.

[30] The ICSI certificate notary, 2015, `https://notary.icsi.berkeley.edu/`.

[31] M. Kranch and J. Bonneau, Upgrading HTTPS in mid-air: An empirical study of strict transport security and key pinning, NDSS, 2015.

[32] A. Langley, Public key pinning, 2011, `https://www.imperialviolet.org/2011/05/04/pinning.html`.

[33] A. Langley, Enhancing digital certificate security, Google Online Security Blog, January 2013, `http://googleonlinesecurity.blogspot.com/2013/01/enhancing-digital-certificate-security.html`.

[34] A. Langley, Further improving digital certificate security, Google Online Security Blog, December 2013, `http://googleonlinesecurity.blogspot.com/2013/12/further-improving-digital-certificate.html`.

[35] A. Langley, Maintaining digital certificate security, Google Online Security Blog, 2014, `http://googleonlinesecurity.blogspot.com/2014/07/maintaining-digital-certificate-security.html`.

[36] A. Langley, Maintaining digital certificate security, Google Online Security Blog, March 2015, `http://googleonlinesecurity.blogspot.com/2015/03/maintaining-digital-certificate-security.html`.

[37] A. Langley, E. Kasper, and B. Laurie, Certificate Transparency, RFC 6962 (Experimental), June 2013.

[38] J. Leyden, Trustwave admits crafting SSL snooping certificate: Allowing bosses to spy on staff was wrong, says security biz, The Register, 2012, `http://www.theregister.co.uk/2012/02/09/tustwave_disavows_mitm_digital_cert/`.

[39] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, et al., AS relationships, customer cones, and validation, in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 243–256, ACM, 2013.

[40] D. Magoni and J. J. Pansiot, Analysis of the autonomous system network topology, SIGCOMM Comput. Commun. Rev., 31(3), pp. 26–37, July 2001, ISSN 0146-4833.

[41] M. Marlinspike, More tricks for defeating SSL in practice, 2009, dEFCON 17.

[42] M. Marlinspike, Ssl and the future of authenticity, 2011, blackHat USA.

[43] M. Marlinspike, Convergence, 2012, http://convergence.io.

[44] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*, CRC press, 1996.

[45] NetGeo, The Internet geographic database, 2015, `http://www.caida.org/tools/utilities/netgeo/`.

[46] NLANR, The national laboratory for advanced network research, 2006, `http://www.caida.org/projects/nlanr/`.

[47] A. Ornaghi and M. Valleri, Man in the middle attacks: demos, 2003, black Hat USA.

[48] Y. Rekhter, T. Li, and S. Hares, A Border Gateway Protocol 4 (BGP-4), RFC 4271 (Draft Standard), January 2006, updated by RFCs 6286, 6608, 6793, 7606, 7607.

[49] R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, 21(2), pp. 120–126, 1978.

[50] J. Schlyter and P. Hoffman, The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA, 2012.

[51] R. Singel, Law enforcement appliance subverts SSL, Wired News, 2010, `http://www.wired.com/2010/03/packet-forensics/`.

[52] R. Sleevi, C. Evans, and C. Palmer, Public key pinning extension for HTTP, 2015.

[53] G. Slepak, The trouble with certificate transparency, September 2014, `https://blog.okturtles.com/2014/09/the-trouble-with-certificate-transparency/`.

[54] C. Soghoian and S. Stamm, Certified lies: Detecting and defeating government interception attacks against SSL (short paper), in *Financial Cryptography and Data Security*, pp. 250–259, Springer, 2011.

[55] TACK, Trust assertions for certificate keys, http://tack.io.

[56] Routeviews peering status report, Technical report, July 2015, `http://www.routeviews.org/peers/peering-status-by-as.html`.

[57] University of oregon route views project, 2015, `http://www.routeviews.org/`.

[58] VASCO, Diginotar reports security incident, August 2011, `https://www.vasco.com/company/about_vasco/press_room/news_archive/2011/news_diginotar_reports_security_incident.aspx`.

[59] Q. Vohra and E. Chen, BGP Support for Four-octet AS Number Space, RFC 4893 (Proposed Standard), May 2007, obsoleted by RFC 6793.

[60] Q. Vohra and E. Chen, BGP Support for Four-Octet Autonomous System (AS) Number Space, RFC 6793 (Proposed Standard), December 2012.

[61] D. Wendlandt, D. G. Andersen, and A. Perrig, Perspectives: Improving SSH-style host authentication with multi-path probing., in *USENIX Annual Technical Conference*, pp. 321–334, 2008.

APPENDIX A

Snapshot of the MRT Files

```
$VAR1 = \{
            'peers' => [
                         {
                           'peer_ip' => '206.126.236.7',
                           'bgp_id' => '212.161.178.92',
                           'as' => 6730
                         },
                         {
                           'peer_ip' => '206.126.236.10',
                           'bgp_id' => '87.86.76.74',
                           'as' => 4589
                         },
                         {
                           'peer_ip' => '206.126.236.12',
                           'bgp_id' => '129.250.0.183',
                           'as' => 2914
                         },
                         .
                         .
                         .
                         {
                           'peer_ip' => '2001:504:0:2:0:5:9605:1',
                           'bgp_id' => '151.248.97.29',
                           'as' => 59605
                         }
                       ],
            'collector_bgp_id' => '206.126.236.142',
            'subtype' => 1,
            'timestamp' => 1438387202,
            'type' => 13,
            'view_name' => undef
          };

$VAR1 = \{
            'subtype' => 2,
```

```
            'sequence' => 0,
            'timestamp' => 1438387202,
            'entries' => [
                          {
                            'ORIGIN' => 0,
                            'NEXT_HOP' => [
                                            '206.126.236.120'
                                          ],
                            'peer_index' => 18,
                            'AS_PATH' => [
                                           41095,
                                           3356
                                         ],
                            'originated_time' => 1438023559
                          }
                        ],
            'bits' => 0,
            'type' => 13,
            'prefix' => '0.0.0.0'
          };

$VAR1 = \{
            'subtype' => 2,
            'sequence' => 1,
            'timestamp' => 1438387202,
            'entries' => [
                          {
                            'ORIGIN' => 0,
                            'NEXT_HOP' => [
                                            '206.126.236.21'
                                          ],
                            'peer_index' => 19,
                            'AS_PATH' => [
                                           13618,
                                           15169
                                         ],
                            'originated_time' => 1438241218
                          },
                          {
                            'ORIGIN' => 0,
                            'NEXT_HOP' => [
                                            '206.126.236.47'
                                          ],
                            'peer_index' => 12,
                            'AS_PATH' => [
                                           19151,
                                           15169
```

```
                ],
  'COMMUNITY' => [
                   '19151:1000',
                   '19151:61001',
                   '19151:65040'
                 ],
  'MULTI_EXIT_DISC' => 0,
  'originated_time' => 1438023851
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.12'
                ],
  'peer_index' => 6,
  'AS_PATH' => [
                 2914,
                 6453,
                 15169
               ],
  'COMMUNITY' => [
                   '2914:420',
                   '2914:1001',
                   '2914:2000',
                   '2914:3000'
                 ],
  'MULTI_EXIT_DISC' => 0,
  'originated_time' => 1438023824
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.25'
                ],
  'peer_index' => 9,
  'AS_PATH' => [
                 6079,
                 15169
               ],
  'MULTI_EXIT_DISC' => 0,
  'originated_time' => 1438023737
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.76'
                ],
```

```
  'peer_index' => 17,
  'AS_PATH' => [
                 5769,
                 15169
               ],
  'MULTI_EXIT_DISC' => 0,
  'originated_time' => 1438023542
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.21'
                ],
  'peer_index' => 18,
  'AS_PATH' => [
                 41095,
                 15169
               ],
  'originated_time' => 1438023495
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.37'
                ],
  'peer_index' => 11,
  'AS_PATH' => [
                 6939,
                 15169
               ],
  'originated_time' => 1438023456
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.26'
                ],
  'peer_index' => 10,
  'AS_PATH' => [
                 16559,
                 15169
               ],
  'COMMUNITY' => [
                   '16559:400',
                   '16559:65401'
                 ],
  'originated_time' => 1438023392
```

```
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.21'
                ],
  'peer_index' => 5,
  'AS_PATH' => [
                 4589,
                 15169
               ],
  'COMMUNITY' => [
                   '4589:2',
                   '4589:420',
                   '4589:670',
                   '4589:674',
                   '4589:10110'
                 ],
  'originated_time' => 1438023135
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.21'
                ],
  'peer_index' => 21,
  'AS_PATH' => [
                 11039,
                 15169
               ],
  'COMMUNITY' => [
                   '24115:15169'
                 ],
  'originated_time' => 1438023045
},
{
  'ORIGIN' => 0,
  'NEXT_HOP' => [
                  '206.126.236.24'
                ],
  'peer_index' => 8,
  'AS_PATH' => [
                 11666,
                 15169
               ],
  'COMMUNITY' => [
                   '11666:2000',
```

```
                                        '11666:2020'
                                      ],
                    'originated_time' => 1438023117
                  },
                  {
                    'ORIGIN' => 0,
                    'NEXT_HOP' => [
                                    '206.126.236.19'
                                  ],
                    'peer_index' => 7,
                    'AS_PATH' => [
                                    3257,
                                    15169
                                  ],
                    'COMMUNITY' => [
                                    '3257:8093',
                                    '3257:30021',
                                    '3257:50002',
                                    '3257:51100',
                                    '3257:51102'
                                  ],
                    'MULTI_EXIT_DISC' => 0,
                    'originated_time' => 1438022720
                  }
                ],
      'bits' => 24,
      'type' => 13,
      'prefix' => '1.0.0.0'
    };
    .
    .
    .
$VAR1 = \{
      'subtype' => 4,
      'sequence' => 553354,
      'timestamp' => 1438387259,
      'entries' => [
                    {
                      'peer_index' => 32,
                      'AS_PATH' => [
                                      11039,
                                      174,
                                      3356
                                    ],
                      'COMMUNITY' => [
                                      '174:21000',
                                      '174:22013'
```

```
                                    ],
                                  'LOCAL_PREF' => 2,
                                  '174:22013' => undef,
                                  'ORIGIN' => 32,
                                  'NEXT_HOP' => [],
                                  'originated_time' => 1438022684
                                }
                              ],
            'bits' => 64,
            'type' => 13,
            'prefix' => '1900:231c:f02::'
          };

$VAR1 = \{
            'subtype' => 4,
            'sequence' => 553355,
            'timestamp' => 1438387259,
            'entries' => [
                           {
                             'peer_index' => 33,
                             'AS_PATH' => [
                                            11666,
                                            6939
                                          ],
                             'COMMUNITY' => [
                                              '11666:20000',
                                              '11666:20010'
                                            ],
                             'LOCAL_PREF' => 2,
                             '11666:20010' => undef,
                             'ORIGIN' => 32,
                             'NEXT_HOP' => [],
                             'originated_time' => 1438113684
                           },
                           {
                             'peer_index' => 29,
                             'AS_PATH' => [
                                            5769,
                                            6939
                                          ],
                             'LOCAL_PREF' => 2,
                             '11666:20010' => undef,
                             'ORIGIN' => 32,
                             'NEXT_HOP' => [],
                             'originated_time' => 1438023429,
                             'MULTI_EXIT_DISC' => 0
                           },
```

```
{
  'peer_index' => 34,
  'AS_PATH' => [
                 19151,
                 6939
               ],
  '19151:65040' => undef,
  'COMMUNITY' => [
                   '19151:1000',
                   '19151:61001',
                   '19151:65040'
                 ],
  'LOCAL_PREF' => 2,
  'ORIGIN' => 32,
  'NEXT_HOP' => [],
  'originated_time' => 1438023423,
  'MULTI_EXIT_DISC' => 0
},
{
  'ORIGIN' => 32,
  'NEXT_HOP' => [],
  'peer_index' => 35,
  'AS_PATH' => [
                 33437,
                 6939
               ],
  '19151:65040' => undef,
  'originated_time' => 1438023419,
  'LOCAL_PREF' => 2
},
{
  'peer_index' => 24,
  'AS_PATH' => [
                 2914,
                 29208,
                 25248,
                 25192
               ],
  'COMMUNITY' => [
                   '2914:410',
                   '2914:1201',
                   '2914:2202',
                   '2914:3200'
                 ],
  'LOCAL_PREF' => 2,
  'ORIGIN' => 0,
  'NEXT_HOP' => [],
```

```
  '2914:3200' => undef,
  'originated_time' => 1438023405,
  'MULTI_EXIT_DISC' => 299
},
{
  'peer_index' => 30,
  'AS_PATH' => [
                6939
              ],
  'LOCAL_PREF' => 2,
  'ORIGIN' => 32,
  'NEXT_HOP' => [],
  '2914:3200' => undef,
  'originated_time' => 1438023388,
  'MULTI_EXIT_DISC' => 1
},
{
  'peer_index' => 25,
  'AS_PATH' => [
                3257,
                1103,
                1101
              ],
  'COMMUNITY' => [
                  '3257:4000',
                  '3257:8030',
                  '3257:50001',
                  '3257:50110',
                  '3257:53100',
                  '3257:53101'
                ],
  'LOCAL_PREF' => 2,
  'ORIGIN' => 32,
  '3257:53101' => undef,
  'NEXT_HOP' => [],
  'originated_time' => 1438022747,
  'MULTI_EXIT_DISC' => 857
},
{
  '4589:10110' => undef,
  'peer_index' => 28,
  'AS_PATH' => [
                4589,
                6939
              ],
  'COMMUNITY' => [
                  '4589:2',
```

```
                                        '4589:420',
                                        '4589:670',
                                        '4589:674',
                                        '4589:10110'
                                      ],
                    'LOCAL_PREF' => 2,
                    'ORIGIN' => 32,
                    'NEXT_HOP' => [],
                    'originated_time' => 1438023170
                  },
                  {
                    '4589:10110' => undef,
                    'ORIGIN' => 32,
                    'NEXT_HOP' => [],
                    'peer_index' => 32,
                    'AS_PATH' => [
                                   11039,
                                   4901,
                                   11164,
                                   6939
                                 ],
                    'originated_time' => 1438022691,
                    'LOCAL_PREF' => 2
                  }
                ],
  'bits' => 32,
  'type' => 13,
  'prefix' => '2001::'
};
```

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Yüce, Emre
Nationality: Turkish
E-mail: emre [at] emreyuce [dot] com
Website: `http://www.emreyuce.com`

EDUCATION

Degree	Institution	Year of Graduation
M.S.	METU IAM Cryptography	2009
B.S.	METU Mathematics	2007
High School	Gazi Anadolu Lisesi	2002

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2015 –	HAVELSAN A.Ş.	Cyber Security Specialist
2009 – 2015	TÜBİTAK ULAKBİM	Senior Researcher
2008 – 2009	Portakal Teknoloji	Application Developer