Security Through Publicity

Eric Osterweil [*] *eoster@cs.ucla.edu*
Dan Massey [†] *massey@cs.colostate.edu*
Batsukh Tsendjav [†] *batsukh@cs.colostate.edu*
Beichuan Zhang [‡] *bzhang@cs.arizona.edu*
Lixia Zhang [*] *lixia@cs.ucla.edu*

Abstract

Current large-scale authentication and non-repudiation systems offer various security measures, but do not meet the needs of today's Internet-scale applications. Though several designs exist, there have been no significant deployments of Internet-scale security infrastructures. In this paper we propose a novel concept called the *public-space* that makes complete information of digital entities' actions publicly available to every user. It is a structured framework that maintains a large number of entities, their actions, relationships, and histories. Posting such information in public does not endorse the information's correctness, but it does provide users with a quantifiable set of information that enables them to detect faults and make informed security decisions. Combined with traditional cryptographic techniques, the public-space system can support the intrinsic heterogeneity of user security requirements in Internet-scale infrastructures and applications.

1 Introduction

Authentication and non-repudiation are central concepts in computer security and a wide range of authentication systems have been developed. For example, PGP[19] is now commonly used to authenticate email messages, the DNS Security Extensions standards (DNSSEC) [7, 9, 8] add origin authentication into the Domain Name System[15], and both S-BGP[13] and SoBGP[16] have been developed to authenticate the origin of Internet routing announcements[1]. All of these are examples of sound designs that rely on public key cryptography to provide basic authentication primitives. DNSSEC, S-BGP, and SoBGP propose to use a rigorous hierarchical structure to verify whether a public key is valid. However, deploying the rigorous hierarchy in DNSSEC and secure routing has proven to be a challenge and there has been no large-scale deployment of these systems. PGP public keys take a self-organized "web of trust" approach which avoids the prerequisite of a rigorously defined delegation hierarchy, and among these approaches, only PGP is actually used in network operations. Unfortunately the current PGP system provides no provable authentication for any given PGP keys. Motivated by challenges in deploying these systems, this paper proposes a new type of Public Space key Infrastructure (PSKI).

The PSKI is not just another general purpose key infrastructure, it operates by obtaining and authenticating *public information*. We formally define public information as data that 1) should be available to any interested party (and, thus, does not involve user privacy issues) and 2) does not vary depending on who requests it. The first requirement says that public information is intended to be well known, and there is no incentive to restrict access to it. The second requirement says that all queries for a dataset result in the same answer. Examples include DNS entries, origins for Internet prefixes, and public keys associated with users. In each of these cases, the owner of the data gains nothing by restricting access to it and it does not vary depending on who requests it.[2] An example of non-public data is an email message, which represents a private communication, and the PSKI is not intended to authenticate this type of data.

The key principle behind the PSKI is that every action should be made in a public space. This public space concept derives from similar principles to those in other areas of public knowledge. Things that are very public (viewable to all) are subject to enough scrutiny so that it becomes very difficult to subvert them. Stated in another way, public data provides its own authentication. The PSKI creates a framework where individual actions are rigorously accounted for. Acting in public does not guarantee that actions will be correct, but it does provide users with a quantifiable set of information and semantics that enable applications to construct meaningful security mechanisms. In other words, the PSKI tracks user-actions in a very rigorous way, such that it lends itself to manual or potentially automated inspection.

[*]Computer Science Department, UCLA
[†]Computer Science Department, Colorado State University
[‡]Computer Science Department, University of Arizona

[1]These routing systems can provide more than just origin authentication, but origin authentication is the most relevant component to this work.

[2]Note that some more complex uses of DNS will actually vary the answer depending on the perceived location of the resolver. For clarity, we don't consider this type of DNS usage in this paper.

Our approach incorporates some of the positive aspects of a hierarchical PKI. Finding a valid path through the hierarchy in a PKI is a necessary and sufficient condition for authenticating data. Similarly, the PSKI creates a universally agreed upon *public space* that has a cross-signing structure (rather than a hierarchy) and data is valid only if it appears in the public space. However, the public space only guarantees the data is a *complete* copy of what was posted, but does not imply that it is correct. Finding a valid path in the PSKI is a necessary condition for authenticating data, but it is not sufficient.

Our approach incorporates aspects of the web of trust and reputation systems to determine whether data is correct. Some rudimentary rules are applied to filter out obviously bad data such as entries' signatures that don't match or data posted by completely unknown entities. Beyond this, it is intentionally easy to post data because the PSKI offers no guarantee that it is correct. The PSKI is based heavily on the notion that entities will be judged by others based on the actions that they take. This differs from reputation-based systems that use a given formula to provide aggregated reputation metrics by digesting user actions. We believe that in large scale *heterogeneous* systems, different users will necessarily have different criteria for judging others. Instead of attempting a universal reputation metric, the main goal of the PSKI is to provide users access to the recorded history of each entity's actions.

The PSKI also implicitly assumes that making data public is acceptable and does not raise privacy concerns. In a system such as DNSSEC or variants of secure BGP, the data is inherently public. There is no privacy concern when posting the public key belonging to a DNS zone or posting the public key belonging to a particular Internet prefix or Autonomous System. In fact, the objective of these systems is to make this data available to anyone who seeks it. Furthermore, the signatures over and actions by these public keys are also intended to be visible to anyone. In these scenarios, the PSKI raises no privacy concerns by making this data public. But this is not necessarily true for all systems. If making actions public introduces privacy concerns, the advantages of PSKI must be weighed against the loss in privacy.

The rest of this paper is organized as follows: Section 2 briefly reviews related work. In Section 3 we describe the basic threat models that face this system and then a complete description of the PSKI approach is given in Section 4. Section 5 shows how the PSKI model works effectively in the context of our motivating examples, DNSSEC and prefix origin authentication. The approach is not limited to these applications and Section 7 discusses our next steps and the future of extending the PSKI into a complete system.

2 Background

Public Key Infrastructures (PKIs) typically provide third-party vouching for user identities through a hierarchical arrangement. They start with a root certificate authority (CA) that, in turn, delegates its authority explicitly to lower levels. For example, the department of defense (DoD) PKI [17], has 2 levels of CAs, DNSSEC uses a PKI that follows the DNS name hierarchy, and S-BGP uses a PKI that follows the address allocation hierarchy. A rigid delegation hierarchy guarantees the binding of public keys and user or organization identities, but it also makes such PKIs difficult to deploy. This difficulty stems from the fact that it requires tremendous cooperation and commitment from different parties. Furthermore, PKIs do not attest to message veracity or user authorization.

As an alternative to authenticating public keys, the Web of Trust [14] uses self-signed certificates and third-party attestations of those certificates. It is very comprehensive because it takes advantage of the *small-world effect*, and it is easy to deploy because it doesn't require a delegation hierarchy. However, the Web of Trust does not address message veracity or how to determine the trustworthiness of a user. Moreover, PGP key servers [2] help users collect other's public keys, but don't record user/entity actions, lack information completeness, and lack admission control.

Reputation systems ([11], [4]) collect information about user behavior and use feedback mechanisms to rate users, usually with simple numerical values. They can be very useful in revealing how trustworthy users are if everyone agrees to trust each other in the same way, to the same degree, and using the exact same criteria. By leveraging certain economies of scale in their target systems, they can be resilient and quantifiable in regards to Byzantine attacks/failures. However, since trust is often a subjective matter, a "one-size-fits-all" reputation metric can be difficult to construct a priori in Internet-scale, heterogeneous systems.

3 Threat Model

The PSKI is designed to combat security issues that the Internet is both facing today and is expected to face in the near future. It offers a convenient platform for applications to build upon when secure authentication amongst Internet users is a necessity.

The PSKI is designed to address three major types of threats. The first type is spoofed or altered messages. Spoofing can occur when a user's communications get intercepted, fabricated, or misdirected. In such cases, an adversary either captures and alters messages or simply injects false messages. The PSKI's key service can be used to implement cryptographic authentication and data integrity verification in order to defend against this type of threat.

However, authentication can only provide the binding between a message and its originator, it cannot address the ve-

racity of a message. In a large-scale Internet system, authenticated messages from different entities may conflict with each other for many different reasons. There can be adversarial entities that intentionally inject false information, compromised keys (or machines) that are used to inject false messages, and/or computers that are misconfigured or out-of-sync that naively send out incorrect information. Inconsistency is bound to exist in large-scale systems. The second type of threat results when one has authenticated (but conflicting) answers, and this is much more difficult to handle.

As an example, one of PSKI's pilot applications is designed to provide authenticated answers for the originating AS of each BGP prefix. In this context, a user, Alice, may claim that a given prefix P belongs to a specific AS origin O_1. An adversary, Eve, may claim that P belongs to another origin AS O_2. When Bob must decide whether O_1 or O_2 is the true origin for the prefix P, the PSKI offers a means for Bob to evaluate which claim is more trustworthy.

The third type of threat comes from adversaries attempting to disturb the operations of any authentication system we put in place. Today's manually operated CAs can be made robust against abuse, however to meet the authentication needs of Internet-scale applications we must have a self-regulated authentication system. Given the absence of an authenticated name space today[3], an adversary may flood an authentication system by registering too many names, or may change his identity after a bad deed. These issues are discussed further in Section 4.

4 Approach

The PSKI maintains two basic types of data, **Entities** and **Actions**. Entities are referenced by identifiers such as email addresses or DNS zone names. For example, "alice@foo.org" may be associated with one or more public keys. Once the PSKI associates a user with her cryptographic entities (keys), all actions by these entities are recorded. If Alice wishes to sign a document, it is really her entity that is used. Hence forth, users that act through their cryptographic entities will simply be referred to as entities. Actions are committed by entities when they sign information in the PSKI. Because of these signatures, actions are non-repudiable.

The relationships between all entities and actions are represented as a **trust graph**. From Figure 1 one can see that the trust graph is directed, with each vertex representing an entity or a generic action, and each edge representing a causal link (such as an entity vouching for another or signing data). Since each action is associated with a timestamp, the trust graph also records the entire history of an entity's behavior in the public space.

The PSKI's incorporation of entity-cross validation with

[3]The most widely used globally unique identifiers today are email addresses. Here, new names are freely created with neither authentication nor traceability.

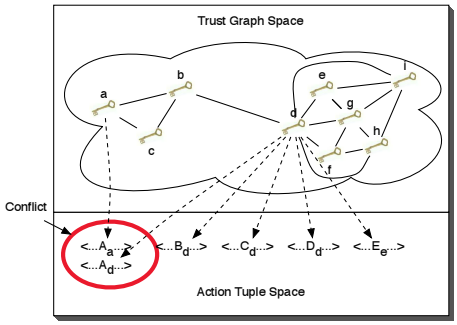

Figure 1: Users with a history of un-contested actions stand out over casual abusers

the actions taken by each creates a synergy that is very powerful. By maintaining a complete public record of entities and their actions over time, users can inspect the actions, detect conflicts, learn about the entities involved, and make their own informed decisions on security issues.

4.1 Entity Management

Entities and the vouching actions among them constitute the *entity space* of the trust graph Each entity in the graph holds the following information:

- The ID (such as an email or a zone name)
- The public key (algorithm/digest/length/etc.)
- The inception and expiration times
- Who signed for this entity to enter the PSKI
- Whom has this entity signed for (i.e. whom has this entity vouched for)
- Lapses in membership
- Emergency rollovers information

Admission: Trust relations cannot be built on sand. To bootstrap a PSKI system, it is necessary to start with a base set of manually verified entities. Each new entity can then be admitted into the PSKI by providing the signatures from at least N entities that are already members of the PSKI, where N serves as a tuning knob to either ease the growth of the system by a small value of N, or reduce (but not eliminate) the chance of admitting adversaries by increasing the value of N. N, therefore, relates to the minimum degree of each node in the trust graph.

An entity in the PSKI is a binding between a name and a key. An immediate question is: does the PSKI attempt to bind a user's physical identity (i.e. Alice Smith) to their entity's identity (alice@foo.org)? The answer is: no, the PSKI does not attempt to verify that an entity correlates to any physical person. Rather, the PSKI simply relates entities to their actions and tolerates the fact that each user may have multiple identifiers, and may wish to bind multiple keys to

each of them. Further, these entities may be vouched for by different sets of entities.

The ease with which a user can abandon their entities raises concern over the possibility that this facility can be exploited for some sort of Sybil attack[6]. For example, an entity might spawn and vouch for several fictitious entities who each cause conflicts. In such a case, the search facility will underscore that the parent entities have a history of vouching for miscreants. This may, in turn, advise end users to distrust those parents and their other children. Furthermore, creating new (conflict-free) entities is self-limiting, in that an entity's lack of history is not likely to impart much trust if a conflict occurs with a more well-known entity. This is particularly true if the existing entity has a conflict-free history.

To build a lightweight, readily deployable, PSKI the authentication power must not depend on external factors such as user traceability in the namespace (i.e. email in PGP). Universal namespace lookup/identification facilities do not exist today and are unlikely to appear in the near future. Instead, the PSKI aims to let each entity authenticate itself through its actions. In other words, trust does not come from a name, but the actions of an entity that establishes its trustworthiness. We explain this concept more in Section 4.2.

It is left to individual users to derive the reputation of entities from their history in the public-space and their relationships to other entities in the trust graph. That said, the PSKI offers data at a low enough level of detail so as be informative in this, but high-level enough to avoid excessive noise.

Membership: Entities that intend to remain active participants in the PSKI are expected to renew themselves in the PSKI when their lifetime expires. This action will be accomplished through a rollover mechanism that enables entities to re-enlist. Failure to do so indicates a lapse in validity, and is tracked in the public-space. It is left to end users to decide if such a pattern of delinquent behavior constitutes a rationale for distrust. We feel that failure to renew is similar to letting one's car insurance lapse, and that a corresponding level of suspicion may be warranted for such behavior in the PSKI. Furthermore, entities that have a history of signing for other entities who display erratic patterns of renewal, or frequently disappear may *themselves* seem suspicious.

Emergency Rollover: In addition to membership maintenance, a mechanism will exist for emergency entity rollover. Such a mechanism can be useful when a key has been lost, compromised, or simply revoked. Naturally, such an action is part of the public-space, and if it is frequently invoked, it my tend to indicate a lesser degree of trust is in order.

4.2 Action Management

Each action that is taken is defined in an application specific way[4], but produces a generic 6-element action-tuple:

$\langle lookup\_key, entity, action, Date_{incep}, Date_{exp}, target \rangle$

An example of this is a signed prefix/origin record (from the BGP Origins application). An action tuple in the BGP Origins application may look like: $\langle 10.0.0.0/8, alice@foo.org, owns, t_1, t_2, ... \rangle$. This record provides a unique search key ($10.0.0.0/8$) that is bound to all signed instances. Each signed instance is then bound to the entity that created it. Furthermore, querying for a prefix may return multiple tuples that claim ownership. In such cases, the conflict resolution can begin by inspecting the entities involved.

Each entity's public-space actions are tracked regarding:

- How often it has signed for items
- How many active (unexpired) signatures it has
- What are those signatures (i.e. a search facility)
- Current conflicts involving this entity
- Total number of conflicts involving this entity

Reputations and accountability: In the PSKI, conflicts between different entities (which are caused by competing signatures) are expected. The PSKI follows a very simple notion of accountability; it does nothing more than strictly report the actions of individual entities. The PSKI offers users the ability to view conflicts, and use the history of each entity to judge the validity of their claims. Moreover, the users are able to traverse the public-space in an attempt to cast further light on each entity. As an example, if two entities each sign for ownership of the same DNS zone in the DNSSEC hierarchy, the conflict can be noted in the PSKI. A user that might want to investigate this conflict may query the PSKI for: any revocation messages that correspond to either entity, the lifetime of each entity (have either expired already), the parent of each entity (were they both vouched for by the same/valid parent), does either entity have a history of conflicts, etc.

The PSKI will attempt to be somewhat agnostic about where keys come from. This attempt is made so that the PSKI can easily be backward compatible with existing PKIs, and remain interoperable with related ongoing key infrastructures. Early usage of the system is targeted toward two particular applications: DNSSEC and BGP Origins. These projects are described in more detail in Section 5.

5 Discussion

The initial goal of the PSKI work is to learn through experience. The early PSKI systems will be structured around the needs of two pilot applications (DNSSEC and BGP) so that we can learn which semantics should be developed for a generalized PSKI.

[4]The initial work done on the PSKI will allow application-specific logic so that a proper/flexible uniform API can evolve.

DNSSEC: DNSSEC is an Internet-scale PKI that faces a number of operational challenges. These challenges include replay attacks and potentially key hijacks. The complexity of this situation is compounded by the fact that the DNS is composed of a multiplicity of operationally isolated administrative domains, which can be problematic in many ways.

The initial DNSSEC PSKI will extend an ongoing DNSSEC monitoring program ([1]). The integration will be such that entities in the PSKI will correspond to DNSKEYs, the trust graph will be the secure delegation hierarchy, and action tuples will be of the form:

$$\langle domain, zonekey, action, T_{incep}, T_{exp}, RRset \rangle$$

BGP Origins: BGP is a very complex system, and it faces a great many challenges. One of the problems is called prefix hijacking. Another early pilot system for the PSKI is an arbitration mechanism to address the cases where multiple autonomous systems (ASes) claim that they are the origins for a routing prefix.

Entities in this system will be PGP public keys[5], the trust graph will be bootstrapped from a web of trust, and action tuples will be of the form:

$$\langle prefix, email, action, T_{incep}, T_{exp}, Origin \rangle$$

5.1 Integration with Other KDCs

There exist many KDCs in the Internet today. Many of them have been designed for servicing the specific goals of their systems, and one of the PSKI's goals is to be compatible with this type of heterogeneity.

One of the KDCs that the PSKI will integrate with first in its pilot is DNSSEC. In DNSSEC, each secure zone maintains a list of its public keys. These keys are periodically rolled over, according the inception/expiration timestamps associated with the DNSKEY record set. DNSSEC keys are vouched for by the parent zones that are directly above them in the DNS hierarchy. The roots of these trees (i.e. the highest zone in a secure hierarchy that does not have a parent that vouches for it[6]) are known as islands of security, and their PSKI entities will need to be vouched for by a PSKI operational key.

PGP key servers exist in may shapes and forms around the Internet. Their roles, as KDCs, is very useful. The BGP Origins pilot will be the first PSKI to use the signing relationships in these PGP servers to import users into the trust graph. Leveraging existing KDCs gives the PSKI the advantage of benefiting from signing parties that already occur regularly.

[5] In the PSKI, operators at the Network Operations Centers (NOCs) that are announcing prefixes can sign for their announcements. Any conflicts can be settled by examining the operator's personal PGP entity in the PSKI.

[6] Note, DNSSEC is an incrementally deployed system, and it is common to find early adopters that have insecure parents.

6 Related Work

Public corroboration of users' identities, actions, and/or data is not novel in of itself. Many atomic concepts that contribute to the synergy in the PSKI have existed in other systems before.

The Simple Public Key Infrastructure (SPKI) [12] defines digital certificates whose main purpose is the authorization of local users, rather than authentication. It assumes the existence of well-defined authorization information, and it does not address entity reputation metrics or misbehaviors.

A recent work [10], called the BBS, uses a notion that is similar to the PSKI, in the context of secure DNS zones. It follows the Web of Trust model and lets zones sign each other's keys to represent trust relationships. The BBS serves as an application specific approach that does not attempt the generality of the PSKI. Moreover, the BBS uses information that is aggregated at the endpoints (resolvers) to form reputations about observed keys. This is in contrast to the PSKI which offers structured information that facilitates the implementation of systems like the BBS over a common framework.

In Oblivious Key Escrow ([3]), the author discusses a large scale *k out of n* key-escrow system. This approach offers the concept of *shareholders* and gathering consensus before allowing any entity to relinquish escrow information. The general notion that parties may make informed decisions based on public information resembles ideas in the PSKI. However, in [3], the author does not offer a rigorous framework such as the *public-space*.

One of the target applications for the PSKI is a BGP Prefix system. The very high-level problem bears a striking resemblance to Pretty Secure BGP (psBGP [18]), but a number of specific decisions set these 2 systems quite far apart. In psBGP, Autonomous Systems (ASes) are given certificates of authenticity from central CAs: registries such as ICANN, IANA, and regional registries (RIRs). This notion, by itself requires significant consideration[7]. They, then, sign their own prefixes and organize themselves into cryptographic peering relationships. Each AS signs its prefixes and signs its neighbors. This hybrid of traditional PKI and peer-to-peer cross signing resembles DNSSEC's key signing and zone signing keys (KSKs and ZSKs). However, unlike the PSKI, psBGP is very specific to BGP, requires *online* cryptographic authentication and processing of BGP data, and requires substantial protocol modifications to BGP. The PSKI's approach is to allow a general *offline* arbitration framework.

Message Authentication by Integrity with Public Corroboration [5] uses notions of offline and independent data sets to verify information seen. This is similar to both the PSKI (in that it uses more than simple cryptography) and to [3] (in that it allows for external venues to add information to a user's decision process). Unlike the PSKI, however, the ex-

[7] Issues such as key rollover have plagues other large scale PKIs (like DNSSEC) and are not addressed in psBGP

ternal mechanisms are not necessarily rigorously specified. [5] offers flexibility, but diminished security.

7 Moving Forward

Today's fault pervasive environment makes it imperative to enhance Internet systems and applications with cryptographic protection. Although there exist solutions today for authentication, verification, and reputation in distributed systems, they fail to support the generality needed by applications and systems that operate in the large-scale and heterogeneous environment of the Internet.

In this paper we have sketched out a novel public space key infrastructure, PSKI. The PSKI design is based on two new ideas: 1) tying crypto entities to their actions; and 2) instead of a reputation index, providing access to recorded history of entity actions, effectively making all entities act in public. Instead of making assumptions about an entity's integrity, acting in public provides the PSKI an effective mechanism to offer integrity checking of individual entities. We believe that the resulting system can support the intrinsic heterogeneity of user security requirements in large scale systems, and provide users a foundation to construct rigorous and quantifiable security protections on top of it.

A number of new challenges exist that must be resolved in order to build a functional PSKI. The system must be distributed to be robust and scalable, it must be tamper-proof, and it must tie entities to their actions. The question of how to build a generic PSKI for multiple applications remains open. However, despite all these unknowns, we believe that the new direction taken by PSKI has a great potential and is worth pursuing. We expect to gain further insight from the two pilot applications.

References

[1] Secspider: A dnssec monitoring tool. http://secspider.cs.ucla.edu/secspider/.

[2] The worldwide public repository for open pgp keys. http://www.keyserver.net/.

[3] Matt Blaze. Oblivious Key Escrow. pages 335–343.

[4] Sonja Buchegger and Jean-Yves Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *P2PEcon 2004*, 2004.

[5] With Public Corroboration. Message authentication by integrity.

[6] John R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.

[7] R. Arends et al. DNS Security Introduction and Requirements. RFC 4033, The Internet Society, March 2005.

[8] R. Arends et al. Protocol Modifications for the DNS Security Extensions. RFC 4035, The Internet Society, March 2005.

[9] R. Arends et al. Resource Records for the DNS Security Extensions. RFC 4034, The Internet Society, March 2005.

[10] Kim Eunjong, Ashish Gupta, Batsukh Tsendjav, and Dan Massey. Resolving islands of security problem for dnssec. In *IWCMC '06: International Wireless Communications and Mobile Computing Conference*. ACM Press, 2006.

[11] Minaxi Gupta, Paul Judge, and Mostafa Ammar. A reputation system for peer-to-peer networks. In *NOSSDAV*, pages 144–152, New York, NY, USA, 2003. ACM Press.

[12] Joseph Y. Halpern and Ron van der Meyden. A logical reconstruction of spki. *J. Comput. Secur.*, 11(4):581–613, 2004.

[13] Stephen Kent, Charles Lynn, and Karen Seo. Secure border gateway protocol (S-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, 2000.

[14] Rohit Khare and Adam Rifkin. Weaving a web of trust. *World Wide Web J.*, 2(3):77–112, 1997.

[15] P. Mockapetris. Domain Names - Concepts and Facilities. RFC 1034, Network Working Group, November 1987.

[16] James Ng. Extensions to BGP to Support Secure Origin BGP (soBGP). Internet draft, Network WG, April 2004.

[17] Rebecca Nielsen and Booz Allen Hamilton. Observations from the deployment of a large scale pki. In *4th Annual PKI R&D Workshop: Multiple Paths to Trust*, April 2005.

[18] Tao Wan, Evangelos Kranakis, and Paul C. van Oorschot. Pretty secure bgp, psbgp. In *NDSS*, 2005.

[19] Philip R. Zimmermann. *The official PGP user's guide*. MIT Press, Cambridge, MA, USA, 1995.