A preliminary version of this paper appears in *ACM Conference on Computer and Communications Security 2012*, pages 541–552, 2012.

Provable Security of S-BGP and other Path Vector Protocols: Model, Analysis and Extensions

Alexandra Boldyreva and Robert Lychev

Georgia Institute of Technology, Atlanta, GA, USA
{sasha,robert.lychev}@gatech.edu

Abstract

This paper provides the provable-security treatment of path vector routing protocols. We first design a security definition for routing path vector protocols by studying, generalizing, and formalizing numerous known threats. Our model incorporates three major security goals. It is quite strong, yet simple to use. We prove by reduction that S-BGP satisfies two out of the security model's three goals, assuming the underlying signature scheme is secure. Under the same assumption, we next show how the protocol can be modified to meet all three security goals simultaneously. We also analyze SoBGP and show that it fails to meet two security goals. Finally, we study security of partial PKI deployment of path vector protocols when not all nodes have public keys. We investigate the possibilities of relaxing the PKI requirement and relying on non-cryptographic physical security of networks that use the protocol in order to achieve possibly weaker, but still well-defined, notions of security. We also present the necessary and sufficient conditions to achieve full security in the partial PKI deployment scenario. We believe our conclusions will prove useful for protocol developers, standards bodies and government agencies.

Contents

1 Introduction

1.1 Motivation and related work

The Border Gateway Protocol (BGP) is currently the de facto standard for routing across the Internet. Its current version, version 4, is defined in a draft standard [49] and is in wide use. In the protocol, each router associated with a particular autonomous system (AS)—an independent network managed by a single administrative entity such as a Content Provider or an Internet Service Provider (ISP)—maintains a list of possible paths to various IP prefixes. AS's advertise information about serviceable prefixes to neighboring AS's who propagate it to their neighbors, and so on, so that reachability information is updated globally.

BGP was designed to enable routing between parties that trust each other and thus it lacks security features. Nowadays, however, commercial interests invalidate the assumption of trust on the Internet. Accordingly, the security of BGP has come under much scrutiny [23, 43, 13, 45] because honest failures or malicious router compromises may cause serious problems throughout the Internet. For example, on April 25, 1997 an incorrect route map was injected into the Internet forcing most Internet traffic to be routed to a small Internet Service Provider (ISP) in Virginia, crippling much of the Internet for about two hours [14]. Similar misconfigurations have been recently documented for Pakistan and China [22, 27], as we will discuss later in the paper. There is a widespread agreement that due to increased importance of the Internet, it is extremely important to ensure security of its infrastructure. The United States Department of Homeland Security views BGP security as part of the national strategy for securing the Internet [46].

Vast related research, including [35, 55, 8, 37, 57, 54, 24, 25], incorporate additional measures to handle authenticity/ integrity and authorization issues in BGP. In particular, a major security vulnerability, such as lack of integrity of the route announcements, has been addressed. Secure BGP (S-BGP) protocol [38, 40] stands out as the most comprehensive attempt to secure the Internet's routing infrastructure to date. It is currently under consideration for standardization by the Internet Engineering Task Force (IETF) [42].

Current security proposals for S-BGP rely on the use of the public key infrastructure (PKI) [6, 41], each party holding a public-secret key pair and a digital certificate on the public key issued by a Certification Authority (CA). To ensure proper integrity/authentication and authorization verification S-BGP utilizes public-key cryptography tools such as digital signatures and their variants.

However, most existing proposals and analyses do not go further than pointing out specific attacks and suggesting possible fixes. For example, although a survey of BGP security [23] throughly discusses such threats as message tampering, session termination, prefix hijacking, prefix deaggregation, subversion of path information, route flapping, etc., it is not immediately clear what precisely an adversary is allowed or supposed to do when attacking BGP. Can it peek on communication, corrupt nodes, collude, etc.? And what are its goals? Even though the proposed solutions may seem plausible, there is no provable way of quantifying their security guarantees. For example, the proposal for secure path vector routing described in [37] without provable security analysis was later shown to suffer from attacks that could be mounted by 60% of AS's on the Internet in [44]. Although this vulnerability was mentioned in [37], there was no way of formally quantifying its seriousness. What is missing is the provable security analysis, which is the superior alternative to the cyclical trial-and-error approach, because the former provides concrete security guarantees. It is a must in modern cryptographic research and design, and it is more and more often required by

the standards bodies.

The only attempt to use provable security (to the best of our knowledge) in the context of securing BGP has been done in [24] (Appendix A). However, there are no details of the security model[1] (it is not clear who is given what keys), the model is very weak: collusions are not addressed (the adversary can only corrupt one AS), route validity—when a route does not contain edges that do not physically exist in the network and no node's export policy on that route is violated—is not captured, and there are no proofs of security. Providing proper provable security treatment for routing protocols is the main focus of our work.

It may be debatable how possible is widespread, near-future adoption and deployment of S-BGP. The main technical reason is that securing BGP adds time and space complexity overhead. There are also political and economic factors, including the financial cost of secure routing. Finally, there is the problem of gradual deployment; that is the necessity of bypassing the impossibility of an instantaneous global change of configurations. But as a position paper on the subject [17] notices, many objections are inherent to any possible solution and are unavoidable. This should not by any means give specialists reasons to stop working on existing problems to make deployable secure BGP a reality, especially given the growing importance of the Internet and its security.

Moreover, several recent efforts in this field justify an optimistic view on S-BGP's deployment [9, 41, 31]. Resource Public Key Infrastructure (RPKI) [5, 41] is a major, real-life, current effort by the Secure Inter-Domain Routing (SIDR) group of the IETF [4] to protect and verify the association of AS numbers and IP prefixes to their owners via cryptographic certificates. Intuitively, RPKI is like an implementation of S-BGP that addresses only a fraction of the attacks that S-BGP is supposed to address (i.e. prefix-hijacking attacks). Results in [31] suggest that the majority of the Internet would deploy secure routing protocols, such as S-BGP, if AS's were to prefer secure routes (routes where every AS deploys S-BGP) to non-secure routes (because by adopting S-BGP, an AS could attract more traffic and increase its revenue). Thus, from an economic point of view, S-BGP could be gradually deployed starting from a small set of AS's.

The above motivation is also applicable to other path vector routing protocols, e.g. BGPSEC [42] and Secure Origin BGP (SoBGP) [55]. While S-BGP is our main focus, many of our results generalize for S-BGP variants and other path vector protocols.

Our work is the first to study path vector routing protocols in the provable-security framework. We formalize the security model for routing protocols and study whether S-BGP meets it. We also address the lack of understanding of its provable security guarantees in scenarios of partial deployment [25]. Studying scenarios that relax the PKI and the public-key-crypto-use requirements for S-BGP, while still achieving reasonable (and well-defined) security levels, is our second main contribution.

Our paper continues a line of work providing provable security treatment for practical protocols, such as SSH [15, 48] and Kerberos [10, 19]. We now describe our contributions in more detail.

1.2 Interdomain network and path vector protocol definitions

We start with defining an interdomain network (a network of AS's such as the Internet) and a path vector routing protocol. Our protocol definition is general enough, and we show how BGP , S-BGP and SoBGP fit. We note that our analysis requires precise notation and definitions; hence we must use some notation and definitions that are not common in the networking literature, but are rather

[1] The details are promised in the technical report, but they do not appear there as well.

standard in the cryptographic literature.

1.3 Security model

Next we design the security definition. We carefully study numerous known security threats and generalize them in a new formal security notion. The definition is strong in that the adversary we consider knows the configuration of the whole network, can observe and modify all communication on the network, can select nodes that will not have public keys (for modeling partial PKI deployment), can corrupt as many AS's as it wants, can learn all secret information of the corrupted AS's, and can act arbitrarily on their behalf. Our model takes into account adaptive corruptions. Our security definition elegantly captures scenarios when not all nodes have public keys (i.e. partial PKI deployment), and, at the same time, it is compact and simple. The adversary is successful if it makes an honest node accept a route announcement that is not legitimate in at least one of the following three ways: (1) unauthentic origin, (2) unauthentic route, (3) invalid route. If no efficient adversary can succeed with noticeable probability in the above three ways, we say that the protocol guarantees full security or (1) origin authentication, (2) route authentication and (3) route validity, respectively.

In Section 5 we explain how the numerous known vulnerabilities are captured by just these three cases. For example, case (1) captures attacks of advertising prefixes that do not belong to the corresponding origins (that were not certified by the certification authority (CA)), also known as the prefix hijacking attacks [11]. Case (2) captures all attacks that include tampering with any announcement made by an honest AS. This includes as a special case a threat known as violation of connection authentication. Case (3) captures somewhat less known attacks on S-BGP such as export policy violations, sometimes known as "route leaks", [33] and announcing a route that cannot physically exist in the network [51]. Our unified definition allows one to analyze full security of a routing protocol or consider security against each of the aforementioned classes of attacks separately.

1.4 S-BGP and SoBGP security analyses

We prove (by reduction) that S-BGP does indeed guarantee origin and route authentication if the utilized building blocks such as a certification and signature schemes are secure (we also prove that a secure certification protocol can be constructed from a secure signature scheme). This formally justifies the design of S-BGP as a means to protect against some of the major threats. However, we also show that S-BGP does not guarantee route validity by presenting explicit attacks under our definition. This is not surprising as it has been shown before, albeit without provable security analysis, and several solutions have been proposed [51, 33, 52, 29]. We propose simple fixes to S-BGP that involve the certification authority certifying links and financial relationships between AS's and we prove that the modified protocol guarantees route validity if the underlying certification protocol is secure. This is somewhat similar to AS policy certificates used in SoBGP [55], except in SoBGP such information is not certified by a third trusted party but by the AS's themselves. Although requiring such certificates may seem inefficient and AS's may be unwilling to make their connections, business relationships and export policies known, we argue that without link-certificates route validity cannot be guaranteed in general. Furthermore, in light of current efforts of RPKI [5] to, even partially, protect approximately 400K currently existing prefixes [47, 2] with cryptographic certificates, we believe that requiring the extra management of cryptographic

link certificates to protect all links, of which there could be over 200K according to recent studies [26, 7], to be still reasonable even though they may require more frequent updates.

SoBGP [55] is another well-known effort to secure BGP. It has already been discussed within the community that SoBGP "should" guarantee origin authentication but that it does not guarantee route authentication and route validity. We use our security model to confirm these conclusions about SoBGP in a provably secure manner. It is not immediately clear whether simple fixes to SoBGP can address these weaknesses, so we focus only on S-BGP when considering partial PKI deployment.

1.5 Relaxing the PKI requirement

Of course, reliance on full PKI deployment and the use of public-key cryptography, while seemingly necessary for strong security, are quite expensive measures. We study the effect on security from having partial PKI deployment, i.e. when not all nodes have certified public keys, and put forth results that can facilitate our understanding of how gradual deployment (and even full deployment, but where, for efficiency reasons, not all parties want to execute parts of the protocol that require the use of their private keys) of secure routing protocols on the Internet could be made possible. Studying security of the partial PKI deployment of path vector protocols is our second main contribution, and the results here are more unexpected and technical.

We first show that S-BGP fails to provide route authenticity if there is at least one node without a certified public key. However, we show that the loss of PKI-related security can be compensated by exploiting physical security of links together with a trust relationship that neighboring nodes must have to establish a physical communication link between them in the first place, and we show that full security is possible if nodes do not select routes with more than one keyless node in a row at any part of those routes. We then show that such restrictions are in fact necessary. Finally, we show that if all prefixes and links are certified by a trusted certification authority, even when no node has a public key, nodes are guaranteed to discover valid routes with authentic origins, and the worst thing that can happen is that an honest node may accept a route to some prefix such that for at least one honest node on that route, the latter does not prefer its part of that route the most. We then argue that in this setting, due the Internet's lack of any provably secure accountability mechanism, the Internet as a whole is just as protected against adversaries whose primary goal is to divert traffic onto unwanted routes as when PKI is fully deployed. Although requiring link certificates while not requiring full PKI deployment may seem to have limited practical gains, this result is a major leap toward understanding the security guarantees and efficiency tradeoffs that can be achieved even when no node has a public key. This result suggests that in the initial stages of partial deployment of secure path vector protocols, it may be more beneficial to deploy link certificates rather than have some nodes possess public keys while deploying no link certificates. We discuss this further with respect to partial and full deployment of RPKI on the Internet.

2 Preliminaries

NOTATION AND CONVENTIONS. We denote by $\{0,1\}^*$ the set of all binary strings of finite length. If x, y are strings then (x, y) denotes the concatenation of x and y from which x and y are uniquely decodable. If $\kappa \in \mathbb{N}$ then 1^κ denotes the string consisting of κ consecutive "1" bits. If S is a finite set, then $s \xleftarrow{\$} \mathsf{S}$ denotes that s is selected uniformly at random from S. If $\mathcal{A}$ is a randomized

algorithm and $n \in \mathbb{N}$, then $a \xleftarrow{\$} \mathcal{A}(i_1, i_2, \ldots, i_n)$ denotes that a is assigned the outcome of the experiment of running $\mathcal{A}$ on inputs $i_1, i_2, \ldots, i_n$. The empty string is denoted by ε. An adversary is an algorithm. By convention, the running-time of an adversary includes that of its overlying experiment. All algorithms are assumed to be randomized and efficient (i.e. polynomial in the size of the input).

2.1 Provable Security Approach

In this work we apply the provable security approach. Unlike the cyclic trial-and-error approach to security, this methodology allows us to have protocols, whose security is provably guaranteed, as long as the assumption about the underlying hard problem remains true for computationally bounded adversaries. This approach consists of the following components. (1) A formal definition of a protocol's syntax. (2) A formal definition of the security task in question that includes a precise description of adversarial capabilities and when is the adversary considered successful. (3) A reduction proof showing that the only way to break the protocol according to the definition is by breaking the underlying problem, believed to be hard. Such treatment requires precise notation and definitions at each of the above steps. Hence, we introduce some notation and definitions that were not common in the networking literature, but are rather standard in the cryptographic literature. We provide informal explanations wherever possible to make the formalisms easier to follow.

We note that our work does not follow the alternative formal-methods (symbolic) approach. To the best of our knowledge such analysis has not been done for the routing protocols as it requires some innovations, such as dealing with lists. When done, it will allow for automatic verification, but still will not imply security in the strongest computational model (and our analysis does) as the required soundness theorems are to rely on the unrealistic properties of signatures.

2.2 PKI and Signature Schemes

Whenever we use public keys, we also (implicitly) assume that a *public key infrastructure (PKI)* is supported, i.e. the public keys are valid, bound to users' identities and are publicly known.

A *digital signature scheme* $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ with associated *message space* MsgSp is defined by three algorithms. The randomized *key generation* algorithm Kg takes the security parameter 1^k and outputs a public–secret key pair: $(pk, K) \xleftarrow{\$} \mathsf{Kg}(1^k)$. The (possibly randomized) *signing* algorithm Sign takes the secret key and message $M \in \mathsf{MsgSp}$ and outputs a signature: $\sigma \xleftarrow{\$} \mathsf{Sign}(K, M)$. The deterministic *verification* algorithm Ver takes the public key, a message and a signature and outputs a bit $b \in \{0, 1\}$ indicating whether the signature is deemed valid or not: $b \leftarrow \mathsf{Ver}(pk, M, \sigma)$. For correctness, it is required that for every (pk, K) output by $\mathsf{Kg}(1^k)$ and every $M \in \mathsf{MsgSp}$ we have that $\mathsf{Ver}(pk, M, \mathsf{Sign}(K, M)) = 1$.

The traditional security notion for a scheme $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ considers an experiment $\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(A)$ associated with an adversary A. First, a pair of keys is generated: $(pk, K) \xleftarrow{\$} \mathsf{Kg}(1^k)$. Then A is given pk and the signing oracle, and it has to output a message and a forgery: $(M, \sigma) \xleftarrow{\$} A^{\mathsf{Sign}(K,\cdot)}(pk)$. The adversary wins and the experiment returns 1 iff $\mathsf{Ver}(pk, M, \sigma) = 1$, $M \in \mathsf{MsgSp}$ and A never queried M to $\mathsf{Sign}(K, \cdot)$. We say that $\mathcal{SS}$ is *uf-cma-secure* if $\Pr\left[\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(A) = 1\right]$ is negligible in k for all efficient algorithms A.

2.3 Certification Schemes

To the best of our knowledge, the certification scheme primitive has not been explicitly defined, but it has been considered as parts of other protocols, e.g. certified encryption and digital signature schemes in [18]. (In the application we consider we will involve the certification protocols to certify prefix ownership for the origins (known as address attestation) as well as neighborship of AS's and the type of business relationship neighbors have with each other.)

A two-party *certification protocol* $\mathcal{CP} = (\mathsf{Kg_{CA}}, (\mathsf{CA}, \mathsf{U}), \mathsf{Vercert})$ is defined by a key generation algorithm, a pair of (possibly) interactive randomized algorithms executed between the certification authority and a user (in our case, an AS), and a verification algorithm. The protocol is associated with an ID space IDSp and data space DSp.

- $\mathsf{Kg_{CA}}$ takes the security parameter 1^k and outputs a public-secret key pair (pk_{CA}, K_{CA}) for the CA.
- CA takes as input a secret key K_{CA}, the identity of user $ID \in \mathsf{IDSp}$ and data $D \in \mathsf{DSp}$. A node's ID is the unique AS number given to the AS associated with that node by the Internet Assigned Numbers Authority (IANA) [3], as is done for every AS on the Internet.
- U takes as input the public key pk_{CA}, the identity $ID \in \mathsf{IDSp}$ and data $D \in \mathsf{DSp}$. As result of the interaction, the outputs of both parties are $\perp$, if something went wrong, or (ID, D, cert), where cert is an issued certificate. We write $((ID,\ D, \mathsf{cert}), (ID, D, \mathsf{cert})) \xleftarrow{\$} (\mathsf{CA}(K_{\mathsf{CA}}, ID, D),\ \mathsf{U}(pk_{\mathsf{CA}},\ ID,\ D))$ for the result of an honest interaction.
- $\mathsf{Vercert}$ takes as input $(pk_{\mathsf{CA}}, ID, D, \mathsf{cert})$ and outputs a bit.

The correctness requirement states that for any pair (pk_{CA}, K_{CA}) output by $\mathsf{Kg_{CA}}(1^k)$, any $ID \in \mathsf{IDSp}$ and $D \in \mathsf{DSp}$, the result of certification $((ID,\ D,\ \mathsf{cert}),\ (ID, D, \mathsf{cert})) \xleftarrow{\$} (\mathsf{CA}(K_{\mathsf{CA}}, ID, D),\ \mathsf{U}(pk_{\mathsf{CA}}, ID, D))$ passes verification, i.e. $\mathsf{Vercert}\ (pk_{\mathsf{CA}},\ ID,\ D, \mathsf{cert}) = 1$.

We now define the security of the certification protocol $\mathcal{CP} = (\mathsf{Kg_{CA}}, (\mathsf{CA}, \mathsf{U}), \mathsf{Vercert})$ with $\mathsf{IDSp}, \mathsf{DSp}$. We call the notion *unforgeability under chosen-data attack*. Consider the following experiment $\mathbf{Exp}^{\text{uf-cda}}_{\mathcal{CP}}(A)$ associated with an adversary A.

First, the CA's keys are generated: $(pk_{\mathsf{CA}}, K_{\mathsf{CA}}) \xleftarrow{\$} \mathsf{Kg_{CA}}(1^k)$. A gets pk_{CA} and after that can repeatedly output (ID, D) so that $ID \in \mathsf{IDSp}, D \in \mathsf{DSp}$ and for each such pair participate in $(\mathsf{CA}(K_{\mathsf{CA}}, ID, D),\ A(pk_{\mathsf{CA}},\ ID,\ D))$ on behalf of the user interacting with the CA.

The experiment outputs 1 iff A at some point returns $(ID', D',\ \mathsf{cert}')$ so that $ID' \in \mathsf{IDSp}, D' \in \mathsf{DSp}$, $\mathsf{Vercert}(pk_{\mathsf{CA}},\ ID', D',\ \mathsf{cert}') = 1$ and CA never output $(ID', D', \mathsf{cert}'')$, for any cert''.

We define A's advantage $\mathbf{Adv}^{\text{uf-cda}}_{\mathcal{CP}}(A)$ in this experiment to be $\Pr\left[\mathbf{Exp}^{\text{uf-cda}}_{\mathcal{CP}}(A) = 1\right]$. We say that $\mathcal{CP}$ is *uf-cda-secure* if $\mathbf{Adv}^{\text{uf-cda}}_{\mathcal{CP}}(A)$ is negligible in k for all efficient algorithms A. Note that one could define a stronger security notion, but that would be an overkill for the purposes of our application.

Construction 2.1. *Let* $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ *be a signature scheme with* MsgSp. *We define the corresponding* $\mathcal{CP}_s = (\mathsf{Kg}, (\mathsf{CA}, \mathsf{U}), \mathsf{Vercert})$ *with* $\mathsf{IDSp}, \mathsf{DSp}$ *so that for every* $ID \in \mathsf{IDSp}$ *and* $D \in \mathsf{DSp}$, $(ID, D) \in \mathsf{MsgSp}$. $(\mathsf{CA}, \mathsf{U})$ *is then as follows. The CA sends* $\mathsf{cert} = \mathsf{Sign}(K_{\mathsf{CA}}, (ID, D))$ *to the user. The user verifies* $\mathsf{Ver}(pk_{\mathsf{CA}}, (ID, D), \mathsf{cert})$ *and, if correct, both output* cert*:* (ID, D, cert)*, otherwise they both output* $\perp$. $\mathsf{Vercert}(pk_{\mathsf{CA}}, ID, D, \mathsf{cert})$ *returns* $\mathsf{Ver}(pk_{\mathsf{CA}}, (ID, D), \mathsf{cert})$.

Theorem 2.2. *Let $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme with message space MsgSp and let $\mathcal{CP}_s = (\mathsf{Kg}, (\mathsf{CA}, \mathsf{U}), \mathsf{Vercert})$ be its corresponding certification scheme with identity and data spaces $\mathsf{IDSp}, \mathsf{DSp}$ as per Construction 2.1. Then, $\mathcal{CP}_s$ is uf-cda-secure if $\mathcal{SS}$ uf-cma-secure.*

Proof. We show that for every adversary A attacking unforgeability of $\mathcal{CP}$, there exists adversary B attacking unforgeability of $\mathcal{SS}$ such that $\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS}}(B) = \mathbf{Adv}^{\text{uf-cda}}_{\mathcal{CP}}(A)$ and the resources of B are that of A.

Let A be an adversary attacking the uf-cda security of $\mathcal{CP}_s$. We construct an adversary B attacking the uf-cma security of $\mathcal{SS}$ as follows. B is given pk_{CA} and the signing oracle $\mathsf{Sign}(K_{\mathsf{CA}}, \cdot)$. For every (ID, D) output by A, B runs (CA, A) with A on behalf of the CA. To compute cert, B queries (ID, D) to its signing oracle and returns the result to A. When A halts and outputs a forgery $(ID', D', \mathsf{cert}')$, B also halts and outputs $((ID', D'), \mathsf{cert}')$.

It is easy to see that the view of A in the simulated experiment has the same distribution as that in $\mathbf{Exp}^{\text{uf-cda}}_{\mathcal{CP}}(A)$ and that B wins, whenever A wins, i.e. B's forgery is valid whenever the same is true for A. Finally, we observe that A and B make the same number of equal-length signing queries and have the same running time. □

3 Interdomain Network Routing

We define syntaxes for interdomain networks and path vector protocols.

3.1 An Interdomain Network

We model an *interdomain network* as a tuple $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$.

- $\mathbf{G}$ is a finite, connected graph consisting of a set of nodes, $\mathsf{AS's}$, that represent autonomous systems and a set of edges, defined by a function $\mathsf{link}\colon \mathsf{AS's} \times \mathsf{AS's} \to \{0,1\}$ that returns 1 iff the nodes are *neighbors*.
- $\mathsf{Prefixes}$ is a set of strings in $\{0,1\}^*$ representing *prefixes*, which specify sets of IP addresses.
- The origin-for-prefix function $\mathsf{OrforPr}\colon \mathsf{Prefixes} \to \mathsf{AS's}$ takes a prefix and returns a node designated to own that prefix (called *origin*).
- $\mathsf{relation}\colon \mathsf{AS's} \times \mathsf{AS's} \to \text{BR}$ is a function that takes two nodes and returns their business relationship if they are neighbors and $\perp$ otherwise [2]. Here BR defines the set of all possible pair-wise business relationships in $\mathcal{I}$ between neighbors. For example, the neighbors could be *peers* or one can be a *provider* and the other its *customer* [30]. Before defining the last components of $\mathcal{I}$, we introduce comments and auxiliary definitions.

  Note that $\mathcal{I}$ implicitly defines the set of origins $\mathsf{Origins} \subseteq \mathsf{AS's}$ as the image set of function $\mathsf{OrforPr}$. We denote the set of neighbors of a node N as $\mathsf{Neighbors}(N)$.

[2] link may be redundant given $\mathsf{relation}$, but we keep the former to maintain a general graph definition

- A *route* in $\mathcal{I}$ is a sequence of nodes $(N_n, N_{n-1}, \ldots, N_2, N_1)$, for some $n \in \mathbb{N}$ and $N_i \in \mathsf{AS's}$ for all $1 \leq i \leq n$, such that $N_1 \in \mathsf{Origins}$. Here N_1 is the destination of traffic and N_i is a possible source of traffic for every $2 \leq i \leq n$. Unless otherwise specified, for convenience, nodes on routes will be indexed in increasing order right-to-left, starting with the origin. We say that N_i is up- or down-stream from node N_j on a particular route, if $i < j$ or $i > j$ respectively.

- A *subroute* of some route $R = (N_n, \ldots N_2, N_1)$ is a sequence of nodes $(N_i, \ldots N_1)$, for any $1 \leq i \leq n$, that is defined as the i right-most entries of R. A route is said to be *feasible* if for every pair of consecutive nodes (N_{i+1}, N_i) in that route, $\mathsf{link}(N_{i+1}, N_i) = 1$ for $n < i \leq 1$, i.e. the nodes are neighbors. A route $(N_n, \ldots N_2, N_1)$ is said to be *to* some prefix $P \in \mathsf{Prefixes}$ if $\mathsf{OrforPr}(P) = N_1$.

- The function $\mathsf{preferto}$ specifies total and transitive binary relations $\mathsf{preferto}_N$ on routes to the same prefix in $\mathsf{Prefixes}$ for each node $N \in \mathsf{AS's}$.

- policy specifies functions policy_N that define export policy rules for each node $N \in \mathsf{AS's}$; policy_N takes a route to some prefix P together with the output of $\mathsf{relation}$ on N and the first node on that route (the second parameter is ignored if N owns P) and outputs a set of nodes to which N is allowed to export (i.e. advertise) that route. With this syntax we consider only next-hop export policy functions whose outputs depend on the routes and business relationships of neighbors on those routes of the node exporting the route, since they are believed to quite reasonably approximate the export policy rules that AS's on the Internet of today use to advertise their routes to different neighbors [30, 33, 31]. We comment on how our results could be extended for more complicated export policy functions in Section 7.

We say that $N_i \in \mathsf{AS's}$ *prefers* some route R to some other route R', both to the same prefix P, if R $\mathsf{preferto}_{N_i}$ R', and we say that a route $R = (N_{n-1}, \ldots, N_2, N_1)$ to prefix $P \in \mathsf{Prefixes}$ is node N_n's j^{th} *most preferred* route to P, for some $j \geq 1$, if there are exactly $j-1$ distinct routes $R' = (M_\ell, \ldots, M_1, N_1)$ to P such that R' $\mathsf{preferto}_{N_n}$ R. We say that R is N_n's most preferred route to P if $j = 1$. For any node N_n, for any route $R = (N_{n-1}, \ldots, N_2, N_1)$ to some prefix P, R $\mathsf{preferto}_{N_n}$ ε iff $\mathsf{OrforPr}(P) = N_1$ unless $\mathsf{OrforPr}(P) = N_n$, in which case ε is N_n's most preferred route to P.

A route $R = (N_n, \ldots N_2, N_1)$ is *valid* if it is feasible and consistent with policy of every node on that route, i.e. $N_i \in \mathsf{policy}_{N_{i-1}}((N_{i-1}, \ldots N_2, N_1), \mathsf{relation}(N_{i-1}, N_{i-2}))$, for all $2 \leq i \leq n$.

Our model of an interdomain network is certainly a simplification of the Internet of today. For example two neighboring AS's could have multiple distinct business relationships at different locations, and any AS's route preference relation and export policy rule could also be a function of the prefix corresponding to the route. However, such details are not necessary to study the essential attacks on the current Internet's routing infrastructure. Furthermore, our network model can be easily extended to incorporate extra features, possibly at the expense of making the analysis more complicated. For instance, one could consider a graph where each node represents a border gateway (router at the border of neighboring AS's), and one could require that the preference relation and the policy function are defined for each prefix.

3.2 A Routing Path Vector Protocol

Let $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$ be an interdomain network. An interactive and stateful *path vector protocol* $\mathcal{PV} = (\mathsf{Init}, \mathsf{An})$ is defined by two algorithms.

- Init is an optional randomized algorithm run by a node (or a CA) that takes the security parameter 1^k and generates the corresponding public and secret keys for the node (or the CA).

- An is a stateful and possibly randomized, interactive multiparty algorithm run between the nodes and possibly the CA. Each node $N \in \mathsf{AS's}$ is given inputs (N, $\mathsf{Neighbors}(N)$, $\mathsf{relation}_N$, $\mathsf{preferto}_N$, policy, $\mathbf{P}_N, pk_{\mathsf{CA}}, \mathbf{pk}$), where $\mathsf{relation}_N$ outputs $\mathsf{relation}(N, N')$ for all $N' \in \mathsf{Neighbors}(N)$ and $\perp$ otherwise. $\mathbf{P}_N \subseteq \mathsf{Prefixes}$ is the set of prefixes N owns, pk_{CA} is the optional public key of the CA and $\mathbf{pk}$ denotes the optional set of public keys of all nodes in $\mathsf{AS's}$. The optional CA takes as inputs $(\mathcal{I}, pk_{\mathsf{CA}})$. During the execution, N_i sends messages known as *route announcements* to $N_j \in \mathsf{Neighbors}(N_i)$, in accordance with policy_{N_i}, of the form (N_i, N_j, R, P, W, Aux), where R is a route to $P \in \mathsf{Prefixes}$ known as the *path attribute*, $W \in \{0,1\}$ is the withdrawal flag, and $Aux \in \{0,1\}^*$ holds any additional information. Upon receipt of a route announcement, N_j can *reject* it by outputting $\perp$. We say that N_j *accepts* a message if N_j does not reject it.

Note that although export policy function of each node is given as input to each node, nodes cannot find out other node's decisions with respect to exporting arbitrary routes, because they are not provided with information in regards to the business relationships of remote nodes and what the feasible routes of remote nodes may be. We comment on how our results could be extended for scenarios when other nodes' export policy rules are not publicly known in Section 7.

We say that $\mathcal{PV}$ is *correct* for a class of networks $\mathcal{C}$ if when every node in $\mathsf{AS's}$ follows $\mathcal{PV}$, every announcement during its execution is accepted for every network $\mathcal{I} \in \mathcal{C}$.

One could consider a stricter notion of correctness that would require path vector protocols to be useful and allow nodes to learn routes to various destinations, e.g. in practice path vector protocols such as BGP are considered useful for the Internet only if they converge. We say that $\mathcal{PV}$ *converges* over $\mathcal{I}$, if after a finite number of sent route announcements every node selects that node's most preferred route, out of all routes it receives as announcements from neighbors, to every prefix that the node has a valid route to in $\mathcal{I}$, such that *subroute consistency* is satisfied. Subroute consistency requires that if $R_i = (N_{i-1}, \ldots, N_1)$ is the most preferred route selected by N_i to P, then for every $1 < j < i$, subroute $R_j = (N_j, \ldots, N_1)$ of R_i is the most preferred route selected by N_j to P, for all $P \in \mathsf{Prefixes}$ and all $N_i \in \mathsf{AS's}$. We say that $\mathcal{PV}$ *diverges* over $\mathcal{I}$ if during its execution there is at least one node in $\mathcal{I}$ that keeps on switching between different routes ad infinitum. If $\mathcal{PV}$ does not diverge, but subroute consistency is not satisfied, we say that $\mathcal{PV}$ neither converges nor diverges but comes to an inconsistent, stable state. This is relevant to our discussion of a particular class of attacks in partial PKI deployment scenarios in Section 8.

The convergence requirement may be unnecessarily complicated and is mostly outside of the scope of this paper, so we do not consider it in the correctness definition of $\mathcal{PV}$ protocols.

4 BGP and S-BGP

In this section we first describe BGP, and then show how S-BGP extends it to incorporate security features. Although in our model we do not require communication to be either concurrent or asynchronous, for the rest of the paper we assume only asynchronous communication as it captures delays and re-ordering ubiquitous in real life scenarios.

4.1 Border Gateway Protocol

We present the essential aspects of the the Border Gateway Protocol (BGP) that is used to establish routes on the Internet of today [30, 33, 31]. Let $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$ be an interdomain network. BGP uses no PKI and no CA, so the optional algorithm Init is never invoked. The An algorithm is as follows.

Every node $N \in \mathsf{AS's}$ maintains state in the form of a table T_N, called the *routing table*, which is initially empty. Each field $T_N[P]$ indexed by a prefix $P \in \mathsf{Prefixes}$, for which $\mathsf{OrforPr}(P) \neq N$, is a list consisting of routes to P that N has received as announcements from neighbors. Each route in $T_N[P]$ is ranked such that $T_N[P][i]$ contains N's i^{th} most preferred route to P.

If the node's input $\mathbf{P}_N$ is nonempty (i.e. $N \in \mathsf{Origins}$), then for every prefix $P \in \mathbf{P}_N$, N sends an announcement $(N, N', (N), P, 0, \varepsilon)$, advertising access to P, to every neighbor $N' \in \mathsf{policy}_N((N), \varepsilon)$.

During BGP's execution, when a node receives an announcement advertising a new route to some prefix, that announcement is ignored if that node is contained in the announced route [3] or if the new route is already contained in that node's routing table to that prefix. Otherwise, that node determines the new route's rank in its routing table to the same prefix, records that route and its rank, and, if necessary, updates the ranks of the other routes to that prefix. If the announced route becomes the most preferred route to that prefix, that node propagates that route to its neighbors in accordance with its export policy rules. If a node receives an announcement that is a notification of a withdrawal of a route (i.e. that route should not to be used by the receiving node) stored in its routing table, then that node deletes that entry from its table and propagates that route's withdrawal to its neighbors in accordance with its export policy rules. We describe BGP more concretely below.

For every route announcement $(N', N, R, P, W, \varepsilon)$ that N receives from neighbor N', if R and $T_N[P]$ do not contain N and R respectively, N sends a route announcement to every neighbor as per policy_N and updates $T_N[P]$ according to rules (1)-(3) below.

(1) If the announcement presents the most preferred route to P, i.e. $W = 0$ and R $\mathsf{preferto}_N$ $T_N[P][1]$, then N:

   (a) sends a route withdrawal announcement (N, N', (N, $T_N[P][1]$), $P, 1, \varepsilon$) to every neighbor as per policy_N, [4]

   (b) sends a route advertisement $(N, N',\ (N, R),\ P, 0, \varepsilon)$ to every neighbor as per policy_N,

   (c) increments by one the rank of every route in $T_N[P]$ and makes an update $T_N[P][1] \leftarrow R$.

(2) If the announcement presents a route to P that is not the most preferred, i.e. $W = 0$ and $T_N[P][1]$ $\mathsf{preferto}_N$ R, then N determines rank i such that R is the i^{th} most preferred route out of all routes in $T_N[P]$, increments by one the rank of every route in $T_N[P]$ that is less preferred than R, and makes an update $T_N[P][i] \leftarrow R$.

(3) If the announcement is a withdrawal of a route that N has stored, i.e. $W = 1$ and $R \in T_N[P]$, then N:

   (a) if $R = T_N[P][1]$, sends a withdrawal announcement $(N,\ N', (N, R),\ P, 1, \varepsilon)$ to every neighbor as per policy_N,

[3] This mechanism is called loop detection, and it is used to prevent routing loops.

[4] Although in practice withdrawals in this specific scenario may be implicit, we make them explicit here for clarity.

(b) if $R = T_N[P][1]$ and $T_N[P][2] \neq \varepsilon$, sends a route advertisement $(N, N', (N, T_N[P][2]), P, 0, \varepsilon)$ to every neighbor as per policy_N,

(c) removes R from $T_N[P]$ and decrements the rank of every route in $T_N[P]$ ranked higher than R.

N ignores new announcements in all other cases. In the absence of adversaries and errors, no message in BGP should be rejected, so BGP should be correct for various interesting classes of networks described in [30, 33, 31] that are believed to closely capture how routing is done on the Internet. Although BGP route announcements in practice may contain more information (that could be stored, for instance, in the *Aux* field) than what we present above, this information is not essential for this paper.

4.2 Secure Border Gateway Protocol

The Secure Border Gateway Protocol (S-BGP) [38] is an extension to BGP that relies on the full deployment of PKI (each AS should know authentic and valid public keys of other AS's). In S-BGP, public-key cryptography is used to bind prefixes to their origins with certificates, called *address attestations*, issued by a third trusted party as well as to generate *route attestations*—certificates generated by intermediate nodes on route announcements they propagate. Route announcement recipients verify the origin of the prefix in that announcement and the certificates of the nodes on the route that announcement has traversed. We present the essential operations of S-BGP more concretely below.

Construction 4.1. *Let* $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$ *be an interdomain network, let* $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ *be a signature scheme with* $\mathsf{MsgSp} = \{0,1\}^*$*, and let* $\mathcal{CP}_s = (\mathsf{Kg_{CA}}, (\mathsf{CA}, \mathsf{U}), \mathsf{Vercert})$ *be the corresponding certification protocol as per Construction 2.1. In S-BGP* $= (\mathsf{Init}, \mathsf{An})$*, as part of* Init *the CA runs* $\mathsf{Kg_{CA}}(1^k)$ *to generate* $(pk_{\mathsf{CA}}, K_{\mathsf{CA}})$ *and each AS runs* $\mathsf{Kg}(1^k)$ *to generate* (pk, K)*.* An *is defined as follows.*

If node N_j*'s input* $\mathbf{P}_{N_j}$ *is nonempty (i.e.* $N_j \in \mathsf{Origins}$*), then for every prefix* $P \in \mathbf{P}_{N_j}$*,* N_j *does the following:*

- *CA and* N_j *interact according to* $(\mathsf{CA}, \mathsf{U})$*,* N_j *being* U*. The input to* U *is* $(pk_{\mathsf{CA}}, N_j, P)$*, the input to* CA *is* $(K_{\mathsf{CA}}, N_j, P)$ *and the outputs of both parties are* (N_j, P, cert)*. Address attestation* $AA^P_{N_j} \equiv \mathsf{cert}$ *is* N_j*'s certificate of ownership of* P*.*

- *Next, for every* $N_i \in \mathsf{policy}(N_j, \epsilon)$*,* N_j *runs* $\mathsf{Sign}(K_{N_j}, (N_i, N_j, P))$ *to produce a route attestation,* $RA^i_{R_j}$*, and sends*
  $(N_j, N_i, R = (N_j), P, 0, Aux = (RA^i_{R_j}, AA^P_{N_j}))$ *to* N_i*; here* R_j *is* R*'s subroute authorized by* N_j *for* N_i *to use and propagate in its own route announcements.*

For every new route announcement $(N_{j-1}, N_j, R = (N_{j-1}, \ldots, N_1), P, W, Aux = (RA^j_{R_{j-1}}, \ldots, RA^2_{R_1}, AA^P_{N_1}))$ *that* N_j *receives from some neighbor* N_{j-1}*,* N_j *first performs address attestation and route attestation verification steps as follows.* N_j *runs* $\mathsf{Vercert}$ $(pk_{\mathsf{CA}}, N_1, P, AA^P_{N_1})$ *and outputs* $\perp$ *if the output of this computation is* 0*. Otherwise,* N_j *runs* $\mathsf{Ver}(pk_{N_i}, (N_{i+1}, \ldots N_1, P), RA^{i+1}_{R_i})$ *for every* $1 \leq i \leq j-1$ *and outputs* $\perp$ *if at least one such computation outputs* 0*. If none of the verification steps above results in* $\perp$*, then* N_j *performs the same operations as* N_j *would do in BGP upon receipt of* $(N_{j-1}, N_j, R, P, W, \varepsilon)$*, as per rules (1)-(3) specified in Section 4.1. Then, for every announcement* $(N_j, N_{j+1}, R', P, W', \varepsilon)$ *that* N_j *would send to* N_{j+1} *in BGP,* N_j *now*

runs $\mathsf{Sign}(K_{N_j}, (N_{j+1}, R', P))$ *to get* $RA^{j+1}_{R'_j}$ *and sends* $(N_j, N_{j+1}, R', P, W', Aux')$ *to* N_{j+1} *instead, where* $R' = (N_j, R)$ *and* $Aux' = (RA^{j+1}_{R'_j}, Aux)$.

If the underlying signature scheme $\mathcal{SS}$ is correct, the execution of S-BGP is the same as that of BGP in terms of how nodes update their routing tables and how they decide which routes to announce to their neighbors. Therefore, S-BGP is correct for the same classes of networks as BGP if the underlying signature scheme $\mathcal{SS}$ used to generate address and route attestations is correct.

5 Routing Protocol Security

In this section we provide a security definition for path vector protocols, show how it captures their security vulnerabilities, and discuss the attacks not captured in our model because they cannot be solved with cryptography.

Intuition for the Formal Security Model. In our model, we do not consider malicious CA's, but we do consider malicious AS's. We consider an adversary which is given the CA's public key and the description of the network $\mathcal{I}$ with at least two nodes. The adversary also specifies which nodes will not have public keys and which nodes it wants to corrupt. The adversary is allowed to adaptively corrupt as many nodes as it wants at any point of its attack. In practice, it is unlikely that a malicious party knows the complete configuration of the network including the relations, and can corrupt as many AS's as it wants, but in the definition we target a very strong adversary. We allow the adversary to corrupt multiple nodes to capture collusion. On the Internet, collusion is certainly a plausible scenario, given that multiple AS's could be managed by a single administration with presence in different geographic locations. The adversary is given all the public and secret keys of the corrupted nodes. We assume that the adversary is stateful, i.e. it can preserve state in between stages. All nodes and the CA can interact: the honest nodes and the CA follow the protocol, while the adversary can act arbitrarily on behalf of the corrupted nodes. It can observe and modify all communication.

The adversary wins if it sends a route announcement to an honest node, the node accepts it and either (1) the prefix in the announcement does not belong to the corresponding origin, (2) there is an honest node on the route that never sent the corresponding announcement for the same prefix, and (3) the route is invalid. The latter includes the possibilities of a non-existing (not-connected) route and a route that does not satisfy the export policies of at least one node on that route.

Path vector protocol security definition. Let $k \in \mathbb{N}$ be the security parameter, $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$ be an interdomain network, of size polynomial in k, such that $|\mathsf{AS's}| \geq 2$, and let $\mathcal{P}V = (\mathsf{Init}, \mathsf{An})$ be a path vector protocol that is correct for $\mathcal{I}$. We define the experiment $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{P}V}(A)$, for $0 \leq m \leq |\mathsf{AS's}|$, involving a stateful adversary A as follows.

Given the description of $\mathcal{I}$, A selects the set $\mathsf{nopubk} \subsetneq \mathsf{AS's}$ of nodes that will not have public keys, such that $|\mathsf{nopubk}| = m$. Then, the public–secret key pairs for the CA and all nodes in $\mathsf{AS's} \setminus \mathsf{nopubk}$ are generated via $\mathsf{Init}(1^k)$. Here and further in the paper $\mathbf{pk}$ denotes the vector of public keys of nodes in $\mathsf{AS's} \setminus \mathsf{nopubk}$ and $\mathbf{pk}[i]$ denotes its i'th component. Given all public keys, A can output the initial sets of corrupted and honest nodes which form a partition of $\mathbf{G}$:
$(\mathsf{Honest}, \mathsf{Corrupted}) \xleftarrow{\$} A(\mathcal{I}, pk_{\mathsf{CA}}, \mathbf{pk})$, so that $\mathsf{Honest} \cup \mathsf{Corrupted} = \mathsf{AS's}$ and $\mathsf{Honest} \cap \mathsf{Corrupted} = \emptyset$.

Next A is given all the secret keys of the corrupted nodes $\{\mathbf{sk}[i]\colon\ \mathbf{sk}[\mathrm{i}]$ belongs to a corrupted node$\}$, and it starts the execution of An on behalf of all nodes in $\mathsf{Corrupted}$ with the CA and also with the nodes in Honest. The CA and the honest nodes follow the protocol legitimately, while the adversary can act arbitrarily. In particular, A is allowed to intercept and modify announcements exchanged between neighboring honest nodes as well as send messages on behalf of any honest node. The adversary is given transcripts of all communication (as it happens). The adversary is also allowed to adaptively corrupt more honest nodes, thereby reducing Honest and increasing $\mathsf{Corrupted}$, as it wishes during this stage of the experiment.

The goal of the adversary is to have an honest node, say $N_\ell \in \mathsf{Honest}$, accept an announcement of the form $(N_{\ell-1}, N_\ell, R = (N_{\ell-1}, \ldots, N_1), P, W, Aux)$, so that at least one of the following conditions is true (the indexing of the nodes on the route is not essential for the definition and is done for simplicity only).

1. *Unauthentic origin:* $\mathsf{OrforPr}(P) \neq N_1$. In this case the experiment outputs 1.
2. *Unauthentic route:* there exists $1 \leq i \leq \ell - 1$ so that $N_i \in \mathsf{Honest}$ and N_i never sent announcement $(N_i,\ N_{i+1}, R' = (N_i, \ldots, N_1), P,\ W', Aux')$ for any W', Aux' to N_{i+1}. In this case the experiment outputs 2.
3. *Invalid route:* R is invalid. In this case the experiment outputs 3.

$\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ returns an output as soon as A wins; if more than one condition above holds, $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ outputs the smallest number. We define A's advantage $\mathbf{Adv}^{\text{sec-rout-m-b}}_{\mathcal{I},\mathcal{PV}}(A)$ in this experiment as $\Pr\left[\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A) = b\right]$, for $b \in \{1, 2, 3\}$.

We define $\mathcal{C}_m^{\mathcal{PV}}$ to be the class of all networks which have m nodes without public keys and for which a path vector protocol $\mathcal{PV}$ is correct, for $m \leq |\mathsf{AS's}|$. $\mathcal{PV}$ guarantees *origin authentication*, *route authentication*, and *route validity* with m-partial deployment (m-PD) for a class of networks $\mathcal{C}_m^{\mathcal{PV}}$, if for every $\mathcal{I} \in \mathcal{C}_m^{\mathcal{PV}}$, for every efficient adversary A, the probability that $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ returns 1, 2 and 3 respectively is negligible in k. $\mathcal{PV}$ is *fully secure* with m-PD for a class of networks $\mathcal{C}_m^{\mathcal{PV}}$, if it guarantees origin authentication, route authentication and route validity with m-PD for $\mathcal{C}_m^{\mathcal{PV}}$, i.e. for every $\mathcal{I} \in \mathcal{C}_m^{\mathcal{PV}}$, for every efficient adversary A, the probability of $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ returning 1, 2 or 3 is negligible in k. When $m = 0$, we omit the suffix 0-PD when qualifying security of protocols.

Remarks. Note that, by definition, although A is allowed to adaptively corrupt as many nodes as it desires at any point of the experiment, A cannot be successful in $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ if it is ever the case that $\mathsf{Honest} < 2$.

We also note that corruptions were handled slightly differently in the proceedings version of the paper [20], where we required A to fix the set of corrupted nodes prior to the sending of any route announcements such that $\mathsf{Honest} \geq 2$. As we explained in [20] that captured adaptive corruptions of all but two nodes. Here, in the full version of the paper we present a more general model that captures adaptive corruptions of all nodes in a non-ambiguous way.

Our model does not consider rogue keys and replay attacks. This is very common as it is known that the standard measures like proofs of possession of secret keys during the key registration [6, 50] and the use of timestamps can be used to provide the additional protection. To address rogue key attacks, we could require the adversary to output the public and secret keys of corrupted users to model the situation where users are required to perform proofs of knowledge of secret keys during

key registration. However, all of our results would still trivially hold in this setting, so we do not complicate our model with this extension since rogue-key attacks are not essential to routing protocols and do not enhance the insights we get about the essential, routing-related attacks on BGP. It may be relevant to investigate whether simpler proofs of possession [50, 18] will suffice, but this is beyond the scope of this paper. We discuss rogue key attacks with respect to RPKI in Section 8.3.

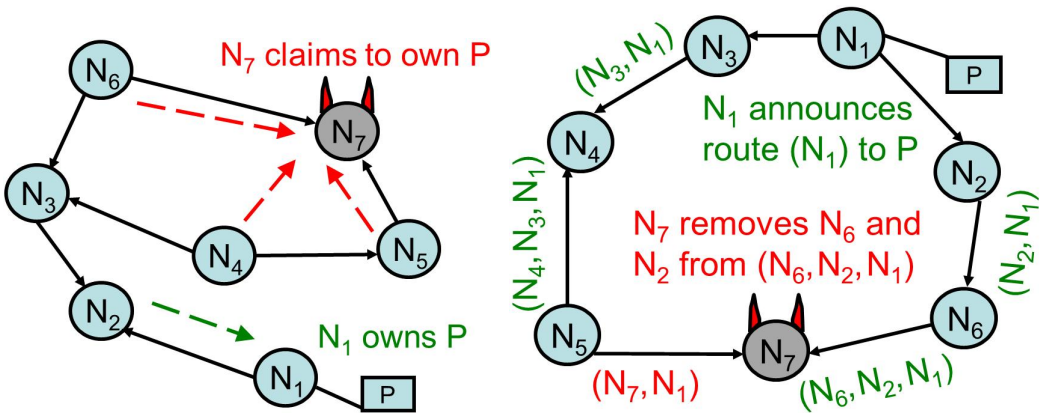
We also note that our security notion does not guarantee that the data that nodes send to those prefixes travels along the routes that they have learned and selected, or whether it reaches those prefixes at all. As shown in [32], path vector protocols cannot guarantee that. These are not goals of path vector protocols, but of data-plane accountability and verification which are outside of the scope of this paper and are not captured in our model.

Although our security model does not take into account all complexities of routing protocols, in Sections 6 -8.3 we show that even a simplified model can point out what is necessary, not just sufficient, to achieve security with respect to essential, fundamental vulnerabilities in path vector protocols in full and partial PKI deployment scenarios .

KNOWN CAPTURED ATTACKS. We discuss how our compact model captures many known vulnerabilities of path vector protocols. For all figures in this section, a directed edge from N to N' indicates that N is N's customer, i.e. N pays N' for all traffic exchanged on their link.

The *Unauthentic origin* condition captures the prefix hijacking attack on BGP, where a corrupt AS claims to own a prefix or announces a more specific prefix, say P, that is owned by another AS. As a result, the corrupt AS could attract potentially all traffic destined to P. With such an attack, a malicious AS could deny access to a particular website, e.g. Pakistan Telecom hijacking YouTube's prefix in February 2008 [22], e.g. by creating a *black hole*—a locale where all traffic destined to P disappears. In addition, the attacker could intercept sensitive, government-related traffic to analyze it for malicious reasons, as speculated by some with regards to China Telecom diverting approximately 15% of Internet's traffic in April, 2010 for about 20 minutes [27]. Prefix deaggration attacks, in which an attacker deaggregates a prefix into more specific prefixes to attract traffic, are also captured by the unauthentic route condition. This works because routers on the Internet select more specific prefixes over less specific ones by default. RPKI [5] is a major, current effort by ARIN [1] to address origin authentication attacks, but by itself RPKI is not intended to address any other types of attacks. Figure 1(a) presents an example of such an attack, where AS N_7 announces to its neighbors ownership of prefix P, whose actual owner is N_1. As a result, N_7 is able to attract traffic from N_4, N_5, and N_6, because N_7 is closer to them than N_1. This traffic never reaches N_1 because, other than through nodes N_5 and N_6, N_7 does not have an alternative route to N_1.

The *Unauthentic route* condition captures known attacks on BGP where an adversarial AS modifies the path attribute of a route announcement by adding and/or taking AS's out of this attribute as well as pretending to be a different AS altogether. By taking AS's out of the path attribute, the attacker could attract more traffic as the advertised route would seem shorter (and thus more preferred). Adding AS's to a route may make a route less attractive if it makes it seem longer, or contains the receiver of the announcement (which would present a loop and cause the receiver to ignore the announcement); this is how an attacker could force an AS not to select certain routes. Figure 1(b) presents an example of such an attack, where AS N_7 removes N_6 and N_2 from the shortest route that N_7 has to P, which is owned by N_1. This makes N_5 believe that N_7 is providing a shorter route to N_1 than the one through N_4, and hence N_5 picks the route through



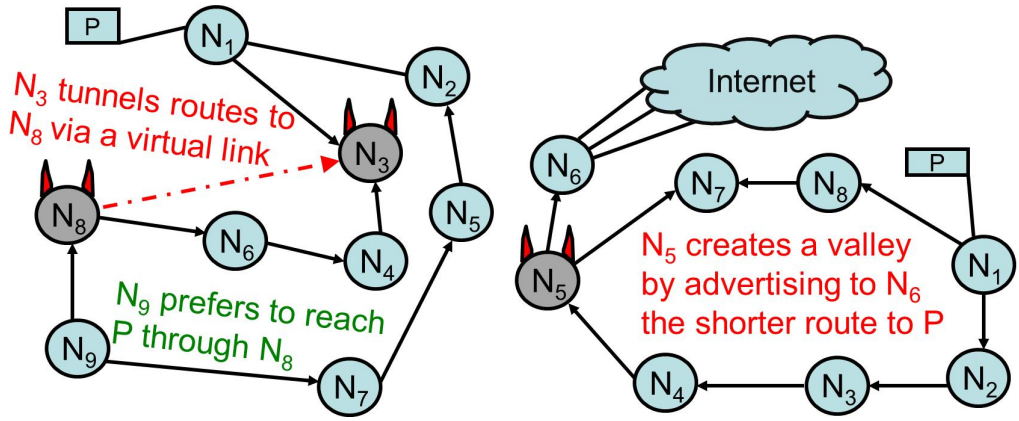
(a) Origin Authentication Attack (b) Route Authentication Attack

Figure 1: In (a) N_7 claims to own prefix P and becomes a black hole by attracting majority of traffic destined to P and dropping it. In (b) N_7 attracts N_5's traffic by advertising a fake short route and then forwarding along a longer route via N_6.

N_7. Thus, N_5 selects a suboptimal route to P, since the route to P through N_7 is actually longer than that through N_4. The attacker benefits not only from intercepting N_5's traffic but also from receiving N_5's payment, since N_5 is N_7's customer.

Connection authentication between adjacent nodes is a special case of route authentication in our security definition. $\mathcal{PV}$ guarantees connection authentication for some network $\mathcal{I}$, whose size is polynomial in k and for which $\mathcal{PV}$ is correct, when the probability of the following event is negligible in k: the adversary A in $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ succeeds in having some honest node N_ℓ accept an announcement of the form $(N_{\ell-1}, N_\ell,\ R = (N_{\ell-1}, \ldots,\ N_1),\ P, W, Aux)$, while $N_{\ell-1}$ is in Honest and has never output announcement $(N_{\ell-1},\ N_\ell, R, P,\ W', Aux')$ to N_ℓ, for any W' and Aux'. This event captures any attack in which an attacker actively modifies route announcements traveling on a link between two non-corrupted nodes and/or impersonates a different AS. A good example of this is the withdrawal attack, where the attacker attempts to make a node withdraw a route that the attacker had never advertised to that node but that was previously advertised to that node by another AS. Since connection authentication is a special case of route authentication, we do not analyze them separately. Furthermore, provable solutions have already been proposed to address them. As suggested in [40], IPSec [39, 28] could be used to prevent attacks on privacy, authenticity and integrity of route announcements exchanged between two non-corrupted nodes.

The *Invalid route* condition captures two known types of attacks on S-BGP, both of which can be used to increase revenue as well as intercept and analyze possibly sensitive traffic. The wormhole attack consists of non-neighboring, colluding AS's attracting traffic by creating a fake (virtual) link between themselves, by tunneling announcements between each other, e.g via IPSec, thereby announcing infeasible routes [56]. (When tunneling announcements, they can essentially *skip* intermediate nodes N_4 and N_6.) Figure 2(a) shows how two AS's, N_3 and N_8, create a fake (virtual) link between each other, although there is no direct path between them. They provide a seemingly shorter route to P, so N_9 selects a route through N_8 and N_3, which is actually longer than the route through N_7 that N_9 would have selected otherwise. The export policy attack consists of an attacker attracting traffic by violating export policy rules. In the example of Figure 2(b), both



(a) Fake Link Attack (b) Export Policy Attack

Figure 2: In (a) colluders N_8 and N_3 create a fake link between each other and attract N_9's traffic. In (b) N_5 attracts traffic of its provider N_6 by violating an export policy rule.

N_6 and N_7 are N_5's providers. By announcing to N_6 the shorter route to P through N_7 instead of the longer route through N_4, N_5 creates a *valley*, i.e. a route through two of its providers, thereby violating a common export policy rule used on the Internet [30]. When forwarding traffic, however, N_5 can use the longer route through its customer N_4, thereby causing N_6 and other nodes in the Internet to use a route longer than they have intended. The same attack can be carried out when either N_6 or N_7 or both are N_5's peers.

Note that in route validity attacks, the adversary introduces routes that are malicious to other users even though they are legitimate from the perspective of S-BGP, i.e. a route that is authentic does not have to be valid. Both types of route validity attacks, the attackers could benefit from intercepting a victim's traffic as well as receiving extra payment from their customers for forwarding it. For networks with more sophisticated export policy rules, more complicated export policy attacks are possible. Route validity attacks have been studied in [51] and [33, 52] respectively, but no provably secure solution has yet been proposed. Also, such situations may be caused by route leaks or non-malicious, unintentional misconfigurations [43, 29] that could still result in responsible AS's suffering from substantial, financial losses.

Attacks Crypto Cannot Prevent. Here we discuss several attacks not captured by our security model for the reason that such attacks cannot be prevented using cryptography.

Path vector protocol divergence cannot be prevented with cryptographic tools since the adversary could keep on withdrawing and then re-announcing the same set of routes ad infinitum. However, since the number of total routes to every prefix is finite, when a protocol diverges, some routes must be periodically withdrawn and then re-announced again (this is called route flapping), so protocol divergence can be mitigated with tools that prevent route-flapping, e.g. route dampening [23]. Convergence of path vector protocols to suboptimal routes, i.e. paths that are not the most preferred, also cannot be prevented with cryptographic tools since the adversary could just make sure that some nodes never receive announcements of the most preferred routes.

Bellovin and Gansner have studied link cutting attacks which involve physically (e.g. with a DDoS attack) taking out edges out of a topology so that certain route announcements fail to propagate [16]. These attacks do not involve the adversary listening and intercepting data without

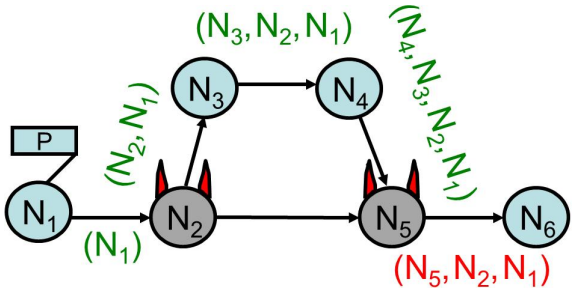


Figure 3: N_5 announces route (N_5, N_2, N_1) to N_6 by signing on behalf of its colluding partner N_2, who never announced route (N_2, N_1) to N_5.

being noticed. Although in our security model the adversary, having access to all communication, can prevent any link from being operational, we do not capture this attack in our security model because, in general, crypto cannot resolve these attacks due to their physical nature.

Finally, contrary to common intuition, path vector protocols cannot guarantee that a particular route announcement was propagated along the route shown in that announcement. Concretely, no path vector protocol $\mathcal{PV}$ can guarantee that for every network $\mathcal{I} \in \mathcal{C}_m^{PV}$, for every efficient adversary A, for any $m \in \mathbb{N}$, the following event occurs with negligible probability in $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$: $N_\ell \in \mathsf{Honest}$ accepts an announcement $(N_{\ell-1}, N_\ell, R = (N_{\ell-1}, \ldots, N_1), P, W, Aux)$ such that there exists $1 \leq i \leq \ell - 1$ so that N_i has never output announcement $(N_i, N_{i+1}, R' = (N_i, \ldots, N_1), P, W', Aux')$ for any W', Aux' to N_{i+1}. Notice that here N_i is not required to be honest as it is in the unauthentic route condition in Section 5.

In Figure 3, we show this attack on S-BGP, where colluding corrupted nodes avoid using their expensive link by sending a route announcement through a path of honest nodes between them, and then taking these honest nodes out of the route announcement. Colluding nodes can do that because they can sign on behalf of each other. In this figure, colluding corrupted nodes N_2 and N_5 avoid using their expensive link (N_2, N_5) by sending an announcement of a route to P through honest nodes N_3 and N_4. After receiving this announcement from N_4, N_5 presents a route (N_5, N_2, N_1) to N_6 by signing on behalf of its colluding partner N_2. N_6 accepts this announcement, even though N_2 has never announced route (N_2, N_1) to N_5. Note that in real-life scenarios, nodes N_2 and N_5 could belong to a single administration with presence in different geographical locations and multiple distinct AS numbers.

6 How Secure is S-BGP?

In this section we show that S-BGP guarantees *origin* and *route authentication*, assuming security of the building blocks, but that it is not fully secure because it does not guarantee *route validity*.

Let $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme, let $\mathcal{CP}_s = (\mathsf{Kg}, (\mathsf{CA}, \mathsf{U}), \mathsf{Vercert})$ be the corresponding straight-forward certification scheme as per Construction 2.1. Theorems 6.1, 6.3 and 6.4 below state our results; the first two are positive and the last one is negative.

Theorem 6.1. *S-BGP per Construction 4.1 guarantees* origin authentication *for* $\mathcal{C}_0^{S\text{-}BGP}$ *if the underlying* $\mathcal{SS}$ *is uf-cma-secure.*

Proof. The proof follows from Theorem 2.2 and Lemma 6.2 stated below. □

In fact, Lemma 6.2 is more general than the above theorem.

Lemma 6.2. *Construction 4.1 guarantees* origin authentication *for $\mathcal{C}_0^{S\text{-}BGP}$ if the underlying $\mathcal{C}P$ is uf-cda-secure.*

Proof. We show that for every adversary A attacking origin authentication of S-BGP, there exists adversary B attacking unforgeability of $\mathcal{C}P$ such that $\mathbf{Adv}_{\mathcal{C}P}^{\text{uf-cda}}(B) = \mathbf{Adv}_{\mathcal{I},\text{S-BGP}}^{\text{sec-rout-m-1}}(A)$ and the resources of B are that of A.

Let A be an efficient adversary attacking origin authentication of S-BGP for a network $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy}) \in \mathcal{C}_0^{\text{S-BGP}}$, with $|\mathsf{AS's}| \geq 2$. We construct an adversary B attacking $\mathcal{C}P$ as follows.

B is given pk_{CA}. B generates (pk_{N_j}, K_{N_j}) for every $N_j \in \mathsf{AS's}$ by running $\mathsf{Kg}(1^k)$. B then gives the description of $\mathcal{I}$ and all public keys to A, and the latter outputs the initial partition $(\mathsf{Honest}, \mathsf{Corrupted})$ of $\mathsf{AS's}$. Next, B gives A all the secret keys of the corrupted nodes, and then A starts the execution of S-BGP on behalf of all nodes in $\mathsf{Corrupted}$ together with B who executes S-BGP on behalf of all nodes in Honest and CA. B follows S-BGP legitimately, whereas A is allowed to act arbitrarily while observing all communication between all nodes in $\mathsf{AS's}$. A is allowed to increase $\mathsf{Corrupted}$ by corrupting more nodes adaptively during its attack.

For each node N_j and prefix P such that N_j owns P (B can check this via $\mathsf{OrforPr}$) and either $N_j \in \mathsf{Honest}$ or $N_j \in \mathsf{Corrupted}$ and A has requested address attestation $AA_{N_j}^P$ of P for N_j, B interacts with the CA via $(\mathsf{CA}(K_{\mathsf{CA}}, N_j, P),\ B(pk_{\mathsf{CA}},\ N_j,\ P))$ to get (N_j, P, AA_j^P). B stores all such certificates $AA_{N_j}^P$. This information together with all honest nodes' secret keys, allows B to follow the computations according to the interactive algorithm An.

Whenever $N_\ell \in \mathsf{Honest}$ accepts an announcement $(N_{\ell-1},\ N_\ell, R = (N_{\ell-1},\ \ldots, N_1), P, W, Aux = (RA_{R_{\ell-1}}^{\ell}, \ldots RA_{R_1}^2,\ AA_1^P))$ such that $\mathsf{OrforPr}(P) \neq N_1$, B outputs (N_1, P, AA_1^P).

It is easy to see that A's view in the simulated experiment has the same distribution of that in $\mathbf{Exp}_{\mathcal{I},\text{SBGP}}^{\text{sec-rout-m}}(A)$. Observe that, in accordance with S-BGP, N_ℓ accepts this announcement only if $\mathsf{Vercert}(pk_{\mathsf{CA}},\ N_1, P, AA_{N_1}^P) = 1$, and, since $\mathsf{OrforPr}(P) \neq N_1$, this means that B has not output $(N_1, P, AA_{N_1}^P)$ as a result of running $(\mathsf{CA}(K_{\mathsf{CA}}, N_1, P),\ B(pk_{\mathsf{CA}},\ N_1,\ P))$ before. Thus, $\mathbf{Adv}_{\mathcal{C}P}^{\text{uf-cda}}(B) = \mathbf{Adv}_{\mathcal{I},\text{S-BGP}}^{\text{sec-rout-m-1}}(A)$. Finally, note that B's running time is the same as that of A. □

Theorem 6.3. *S-BGP per Construction 4.1 guarantees* route authentication *for $\mathcal{C}_0^{S\text{-}BGP}$ if the underlying $\mathcal{SS}$ is uf-cma-secure.*

Proof. We show that for every adversary A attacking route authentication of S-BGP, there exists adversary B attacking unforgeability of $\mathcal{SS}$ such that $\mathbf{Adv}_{\mathcal{SS}}^{\text{uf-cma}}(B) = \frac{1}{|\mathsf{AS's}|}\mathbf{Adv}_{\mathcal{I},\text{S-BGP}}^{\text{sec-rout-m-2}}(A)$ and the resources of B are that of A plus some overhead upper bounded by the size of network using S-BGP that A is attacking.

Let A be an efficient adversary attacking route authentication of S-BGP for a network $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy}) \in \mathcal{C}_0^{\text{S-BGP}}$, with $|\mathsf{AS's}| \geq 2$. We construct an adversary B attacking $\mathcal{SS}$ such that

$$\Pr\left[\mathbf{Exp}_{\mathcal{SS}}^{\text{uf-cma}}(B) = 1\right] = \frac{1}{|\mathsf{AS's}|}\mathbf{Adv}_{\mathcal{I},\text{S-BGP}}^{\text{sec-rout-m-2}}(A).$$

B is given a public key pk and the signing oracle $\mathsf{Sign}(K, \cdot)$. Let n be the size of $\mathsf{AS's}$. B first picks a node's index at random $j \xleftarrow{\$} \{1, \ldots, n\}$ for node $N_j \in \mathsf{AS's}$ and then generates public and secret keys for the CA and all nodes except N_j: $(pk_{\mathsf{CA}}, K_{\mathsf{CA}}) \xleftarrow{\$} \mathsf{Kg}_{\mathsf{CA}}(1^k)$, $(\mathbf{pk}[1], \mathbf{sk}[1]), \ldots, (\mathbf{pk}[j-1], \mathbf{sk}[j-1]), (\mathbf{pk}[j+1], \mathbf{sk}[j+1])\mathbf{pk}[n], \mathbf{sk}[n]) \xleftarrow{\$} \mathsf{Kg}(1^k)$.
B sets $\mathbf{pk}[j] \leftarrow pk$.

Next, B gives the description of $\mathcal{I}$ and all public keys to A and the latter outputs its initial partition $(\mathsf{Honest}, \mathsf{Corrupted})$ of $\mathbf{G}$. If $N_j \in \mathsf{Corrupted}$, then B aborts, otherwise B gives A all the secret keys of the corrupted nodes.

Now A starts the execution of S-BGP on behalf of all nodes in $\mathsf{Corrupted}$ together with B who executes S-BGP on behalf of all nodes in Honest and CA. B follows S-BGP legitimately, whereas A can act arbitrarily. B stores all the communication and also provides A with all communication between all nodes. B has all secret keys to simulate the execution of the protocol except for node N_j. Whenever a secret-key operation is required from it, such as a route attestation for route R destined to N_j's neighbor, B invokes its signing oracle to compute a signature on the corresponding data. A is allowed to continue to corrupt more nodes adaptively as it wishes, and B's simulation would change accordingly with the increase of $\mathsf{Corrupted}$ and the decrease of Honest. B aborts if N_j ever becomes a member of $\mathsf{Corrupted}$.

Whenever $N_\ell \in \mathsf{Honest}$ accepts an announcement $(N_{\ell-1}, N_\ell, R = (N_{\ell-1}, \ldots, N_1), P, W, Aux = (RA^{\ell}_{R_{\ell-1}}, \ldots RA^{2}_{R_1}, AA^{P}_{1}))$ such that there exists $1 \leq i \leq \ell - 1$ so that $N_i \in \mathsf{Honest}$ but N_i has never announced $(N_i, N_{i+1}, R' = (N_i, \ldots, N_1), P, W', Aux')$ for arbitrary W', Aux' (we refer to this event by A frames i), B aborts if $N_i \neq N_j$. Otherwise (if $i = j$), B outputs $((N_{i+1}, N_i, \ldots, N_1, P), RA_{R_i})$.

We now analyze B. It is easy to see that if B does not abort, then its simulation for A is perfect, i.e. A's view has the same distribution as that in $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\text{S-BGP}}(A)$.

Observe that, in accordance with S-BGP, N_ℓ accepts such an announcement only if $\mathsf{Ver}(pk, (N_{i+1}, N_i, \ldots, N_1, P), RA_{R_i}) = 1$, so B's forgery is also valid. Similarly, B's message $R'' = (N_{i+1}, N_i, \ldots, N_1, P)$ is "new", i.e. has not been queried to the signing oracle, because $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\text{S-BGP}}(A) = 2$ only if R'' was not part of any announcement by N_i. This is true because, in S-BGP, the ID of the node that is supposed to receive a route announcement is always part of the message that is being signed to produce a route attestation. Therefore, $\Pr\left[\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B) = 1\right] = \frac{1}{|\mathsf{AS's}|}\Pr\left[\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\text{S-BGP}}(A) = 2\right]$ which we justify as follows, using Bayes Rule. Probability that B wins by outputting $((N_{i+1}, N_i, \ldots, N_1, P), RA_{R_i})$, over all $i \in \mathsf{AS's}$, is

$$
\begin{aligned}
\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS}}(B) &= \sum_{i\in \mathsf{AS's}} \frac{1}{|\mathsf{AS's}|}\Pr\left[\, i \notin \mathsf{Corrupted}\,\right]\Pr\left[\, i \notin \mathsf{Corrupted} \mid A \text{ frames } i\,\right] \\
&= \frac{1}{|\mathsf{AS's}|}\sum_{i\in \mathsf{AS's}} \frac{\Pr\left[\, i \notin \mathsf{Corrupted}\,\right]\Pr\left[\, A \text{ frames } i \mid i \notin \mathsf{Corrupted}\,\right]\Pr\left[\, A \text{ frames } i\,\right]}{\Pr\left[\, i \notin \mathsf{Corrupted}\,\right]} \\
&= \frac{1}{|\mathsf{AS's}|}\sum_{i\in \mathsf{AS's}} \Pr\left[\, A \text{ frames } i\,\right]\Pr\left[\, A \text{ frames } i \mid i \notin \mathsf{Corrupted}\,\right] \\
&= \frac{1}{|\mathsf{AS's}|}\sum_{i\in \mathsf{AS's}} \Pr\left[\, A \text{ frames } i\,\right] \\
&= \frac{1}{|\mathsf{AS's}|}\Pr\left[\, \mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\text{S-BGP}}(A) = 2\,\right] \\
&= \frac{1}{|\mathsf{AS's}|}\mathbf{Adv}^{\text{sec-rout-m-2}}_{\mathcal{I},\text{S-BGP}}(A)\ .
\end{aligned}
$$

B is efficient since, to simulate S-BGP, the number of queries it makes to $\mathsf{Sign}(K,\cdot)$ and their length are upper-bounded by the number of queries that A makes and $\mathcal{I}$'s size respectively. □

Theorem 6.4. *S-BGP as defined in Construction 4.1 does not guarantee route validity for* $\mathcal{C}_0^{S\text{-}BGP}$.

The proof formalizes the aforementioned attacks on S-BGP pointed out in [51, 33]. One attack deals with an adversary forging a connection that does not really exist in the network, and the other presents an adversary forging a route that violates the export policy of an intermediate node. Either attack is sufficient to validate Theorem 6.4, we formalize just the former for simplicity. Note that the AS-level graph of the Internet is not a complete graph, so it is definitely vulnerable to this kind of attack.

Proof. We present an efficient adversary A attacking route validity of S-BGP, by succeeding in having an honest node accept an infeasible route, such that $\mathbf{Adv}^{\text{sec-rout-m-3}}_{\mathcal{I},\text{S-BGP}}(A) = 1$.

We present a general attack in which the adversary corrupts two non-neighboring nodes that are on a valid route, obtains the corresponding route announcement from the corrupted node closer to the origin, and then propagates the corresponding route announcement on behalf of the other corrupted node. The route in the latter announcement is infeasible because the corrupted nodes are not neighbors, but there is no way to verify this fact by honest nodes down-stream from the corrupted node that is farther away from the origin.

Consider an arbitrary network $\mathcal{I} \in \mathcal{C}_0^{\text{S-BGP}}$ that has at least one valid route R that contains at least one pair of two non-neighboring nodes N_i and N_j. A is given the description of $\mathcal{I}$ and the public keys of the CA and all nodes. A picks nodes N_i and N_j on a valid route $R = (N_\ell, \ldots, N_1)$, such that $1 < i < j < \ell$, which are not neighbors ($\mathsf{link}(N_i, N_j) = 0$). A initially returns ($\mathsf{Honest} = \mathsf{AS's} \setminus \{N_i, N_j\}, \mathsf{Corrupted} = \{N_i, N_j\}$). A gets the secret keys for the corrupted nodes and begins the execution of the interactive protocol An on their behalf. A follows the protocol honestly. At some point of the protocol's execution, on behalf of N_i, A receives an announcement of R's subroute $\tilde{R}$, $(N_{i-1}, N_i,\ \tilde{R} = (N_{i-1},\ \ldots, N_1), P, 0, Aux = (RA^i_{\tilde{R}_{i-1}}, \ldots, RA^2_{\tilde{R}_1},\ AA^P_1))$ from $N_{i-1} \neq N_i$, where the components of Aux are computed according to S-BGP's description in Section 4.2.

Then, N_j sends the announcement $(N_j,\ N_{j+1}, R' = (N_j,\ N_i, \ldots, N_1), P, 0, Aux' = (RA^{j+1}_{R'_j},\ RA^{j}_{R'_i}, \ldots, RA^{2}_{R'_1},\ AA^{P}_{1}))$ to N_{j+1}. Note that in this announcement nodes $N_{i+1}, \ldots N_{j-1}$ are removed from R, so, since $\mathsf{link}(N_i, N_j) = 0$, R' is infeasible. $N_{j+1} \in \mathsf{Honest}$ will not reject this announcement, because it will pass the verification process according to S-BGP, as all the signatures in Aux' are valid, and there is no way in general for N_{j+1} to verify whether N_i and N_j are neighbors or not.

A is clearly efficient, and, for $m = 0$, $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\text{S-BGP}}(A)$ will return 3 with probability 1, so $\mathbf{Adv}^{\text{sec-rout-m-3}}_{\mathcal{I},\text{S-BGP}}(A) = 1$.

□

We now present a special case of the general attack described in the proof of Theorem 6.4 with the network depicted in Figure 2(a). Note that this network has at least one valid route $R = (N_8, N_6, N_4, N_3, N_1))$. The adversary A is given the description of this network and the public keys of the CA and all nodes. A corrupts two non-neighboring nodes N_3 and N_8 that are on a valid route R. A gets the secret keys for the corrupted nodes and begins the execution of the interactive protocol An on their behalf. A follows the protocol honestly. At some point of the protocol's execution, on behalf of N_3, A receives an announcement from N_1, $(N_1, N_3, \tilde{R} = (N_1), P, 0, Aux = (RA^{3}_{\tilde{R}_1},\ AA^{P}_{1}))$, where the components of Aux are computed according to S-BGP's description in Section 4.2. Then, on behalf of N_8, A sends the announcement $(N_8,\ N_9, R' = (N_8, N_3, N_1, P, 0, Aux' = (RA^{9}_{R'_8}, RA^{8}_{R'_3}, RA^{3}_{R'_1},\ AA^{P}_{1}))$ to N_9. Note that in this announcement nodes N_4 and N_6 are removed from R, so, since $\mathsf{link}(N_3, N_8) = 0$, R' is infeasible. Honest node N_9 will not reject this announcement, because the latter will pass the verification process according to S-BGP, because all the signatures in Aux' are valid, and there is no way for N_9 to verify whether N_3 and N_8 are neighbors or not.

7 Fully Secure BGP

To address the attack in the proof of Theorem 6.4, we suggest modification to S-BGP and show that the resulting protocol *provably* guarantees route validity assuming the underlying signature scheme is secure. We argue that this modification is necessary. The modified protocol is fully secure (according to our security definition from Section 5) under the same assumption, so we call it *fully secure* BGP or FS-BGP.

Construction 7.1. *Let* $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}),\ \mathsf{Prefixes},\ \mathsf{OrforPr},\ \mathsf{relation},\ \mathsf{preferto}, \mathsf{policy})$ *be an interdomain network, let* $\mathcal{SS} = (\mathsf{Kg},\ \mathsf{Sign}, \mathsf{Ver})$ *be a signature scheme, and let* $\mathcal{CP}_s = (\mathsf{Kg}_{\mathsf{CA}},\ (\mathsf{CA}, \mathsf{U}), \mathsf{Vercert})$ *be the corresponding certification protocol as per Construction 2.1. Let S-BGP =* $(\mathsf{Init}, \mathsf{An})$ *be the construction from Section 4.2. FS-BGP =* $(\mathsf{Init}, \mathsf{An}')$ *is defined exactly like S-BGP, but* An' *requires a few extra operations.*

After all address attestations are generated and before any announcement is sent, each node N_j *interacts with the CA via* $(\mathsf{CA}, \mathsf{U})$. *In what follows, smaller input is always on the left corresponding to any link* (N_j, N_i), *and for convenience only, suppose that* $N_j = \min(N_j, N_i)$, *for every* $N_i \in \mathsf{Neighbors}(N_j)$. *For this interaction, the input to* U *is* $(pk_{\mathsf{CA}},\ N_j,\ ((N_j,\ N_i),\ \mathsf{relation}(N_j, N_i)))$, *the input to* CA *is* $(K_{\mathsf{CA}},\ N_j,\ ((N_j,\ N_i),\ \mathsf{relation}(N_j, N_i)))$ *and the outputs of both parties are* $(N_j,\ ((N_j,\ N_i),\ \mathsf{relation}(N_j, N_i)),\ \mathsf{cert})$. *We define* link attestation *to be* $LA_{N_jN_i} \equiv \mathsf{cert}$. *If* N_j *owns prefix* $P \in \mathsf{Prefixes}$, *for every* $N_i \in \mathsf{policy}_{N_j}((N_j), \varepsilon)$, N_j *generates a route attestation* $RA^{i}_{R_j}$ *just as in S-BGP and sends* $(N_j, N_i,\ R = (N_j), P, 0, Aux = ((\mathsf{relation}(N_j, N_i), LA_{N_jN_i}),\ RA^{i}_{R_j},\ AA^{P}_{N_j}))$ *to* N_i.

For every new route announcement $(N_{j-1}, N_j, R = (N_{j-1}, \ldots, N_1), P, W, Aux = (\mathsf{relation}(N_{j-1}, N_j), LA_{N_{j-1}N_j}, RA^{j}_{R_{j-1}}, \ldots \mathsf{relation}(N_1, N_2), LA_{N_1N_2}, RA^{2}_{R_1}, AA^{P}_{N_1}))$ *that* N_j *receives,* N_j *first performs address and route attestation verification just as in S-BGP, and, if these steps do not result in* $\perp$*, then* N_j *performs link attestation verification as follows.* N_j *runs* $\mathsf{Vercert}$ $(pk_{\mathsf{CA}}, N_i, ((N_i, N_{i+1}), \mathsf{relation}(N_i, N_{i+1})), LA_{N_iN_{i+1}})$*, for every* $1 \leq i \leq j-1$*, and outputs* $\perp$ *if at least one such computation outputs* 0*. Otherwise,* N_j *outputs* $\perp$ *if there is at least one* N_i*, for* $1 \leq i \leq j-1$*, such that* $N_{i+1} \notin \mathsf{policy}_{N_i}((N_i, \ldots, N_1), \mathsf{relation}(N_i, N_{i-1}))$.

If none of the verification steps above results in $\perp$*, then* N_j *performs the same operations as* N_j *would do in S-BGP upon receipt of* $(N_{j-1}, N_j, R, P, W, RA^{j}_{R_{j-1}}, \ldots RA^{2}_{R_1}, AA^{P}_{N_1}))$*, and then, for every message* $(N_j, N_{j+1}, R', P, W', Aux')$ *that* N_j *would send to* N_{j+1} *in S-BGP,* N_j *now sends* $(N_j, N_{j+1}, R', P, W', Aux'')$ *to* N_{j+1} *instead, where* $R' = (N_j, R)$ *and* $Aux'' = (\mathsf{relation}(N_j, N_{j+1}), LA_{N_jN_{j+1}}, RA^{j+1}_{R'_j}, Aux)$.

Note that FS-BGP is correct for the same classes of networks that BGP is correct for, if the underlying signature scheme $\mathcal{SS}$ used to generate address, route attestations and link attestations is correct.

Theorem 7.2. *FS-BGP as defined in Construction 7.1 is fully secure for* $\mathcal{C}_0^{FS\text{-}BGP}$ *if the underlying* $\mathcal{SS}$ *is uf-cma.*

Proof. The proof follows from Theorems 2.2, 6.1, 6.3 and Lemma 7.3 stated below. □

Lemma 7.3. *FS-BGP, as defined above, guarantees route validity for any network* $\mathcal{I} \in \mathcal{C}_0^{FS\text{-}BGP}$ *if the underlying* $\mathcal{CP}$ *is uf-cda-secure.*

Proof. The proof is very similar to the proof of Lemma 6.2. We show that for every adversary A attacking route validity of S-BGP, there exists adversary B attacking unforgeability of $\mathcal{CP}$ such that $\mathbf{Adv}^{\text{uf-cda}}_{\mathcal{CP}}(B) = \mathbf{Adv}^{\text{sec-rout-m-3}}_{\mathcal{I},\text{S-BGP}}(A)$ and the resources of B are that of A.

Let A be an efficient adversary attacking route validity of S-BGP for a network $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy}) \in \mathcal{C}_0^{\text{S-BGP}}$, with $|\mathsf{AS's}| \geq 2$. We construct an adversary B attacking unforgeability of $\mathcal{CP}$ as follows.

We use CA and CA interchabeably. Given the CA's public key pk_{CA}, B generates the public and secret keys for all AS's, and gives A all public keys, including that of CA. After A outputs the initial sets of honest and corrupted nodes, B interacts with A according to the interactive protocol $\mathsf{An'}$. We observe that the only information that B cannot initially compute are the address attestation certificates for any of the prefixes and link attestation certificates for any of the links. To obtain these certificates, B can sequentially interact with the CA via (CA, B) on the appropriate inputs. Finally, we observe that A's forgery, i.e. an announcement that contains an invalid route that passes the verification, can be converted into B's forgery. This is because an invalid route implies that at least one link attestation in the announcements field Aux contains a valid signature on the data that the CA never signed. This is because an invalid route requires that at least one pair of subsequent nodes on a route advertised in that announcement are not neighbors and/or one of them violated the export policy rule. In our case the policy only depends on the relationships between neighboring nodes. Therefore,

$$\mathbf{Adv}^{\text{uf-cda}}_{\mathcal{CP}}(B) = \mathbf{Adv}^{\text{sec-rout-m-3}}_{\mathcal{I},\text{S-BGP}}(A).$$

Note that B's running time is the same as that of A. □

Assigning link attestations for every link in the Internet may seem impractical because the Internet contains many more edges than AS's (possibly over 200K versus 40K [26, 7]), their management is harder due to periodic reconfiguration, and AS's may be unwilling to expose their connections, business relationships and export policies. However, we argue that link attestations are necessary to prevent route feasibility attacks in general. If a path vector protocol guarantees route validity, every announcement received as part of this protocol can itself serve the role of a certificate for the links between the nodes in the route of that announcement. Since in our model arbitrary nodes on any route could be corrupted, such certificates would have to be generated independently by trusted parties. Analogously, to guarantee route validity when export policies of nodes are not publicly known and/or are not next-hop, more sophisticated certificates and in greater amounts (potentially one for every route of every node, and to every prefix of every origin) would have to be issued by a trusted authority to ensure that honest nodes can check for export policy violations of remote nodes.

Several plausible solutions to route leaks–unintentional export policy violations—and route validity attacks have been suggested without provable security analysis in [52, 29]. Although these solutions are more practical than FS-BGP because they are mostly based on restricted models of AS's business relationships and export policies, e.g. models presented in [30], it is not clear whether they work with respect to colluding adversarial AS's. Also, because business relationships and export policies of AS's on the Internet may be more complicated than in the model of [30], as we argued above, a more sophisticated solution than what the ones proposed in [52, 29] would be necessary.

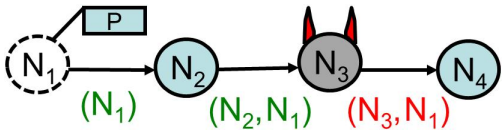
In SoBGP [55], Origin Authorization Certificates are used to bind prefixes to certain AS's (just like address attestations in S-BGP) while AS Policy Certificates are used to allow nodes to learn of links and policies of remote nodes. Although similar to link attestations, these certificates are not generated for links by a third trusted party; instead nodes (possibly corrupted) themselves disseminate their neighborship information. In Appendix A we formally define SoBGP and prove that it guarantees origin authentication but does not guarantee route authentication and route validity. The latter two points can be shown by constructing attacks similar to those in Theorems 8.4 and 6.4 respectively.

8 Partial Deployment of PKI

In this section we study the effect on security of the partial deployment of PKI. We first show that neither S-BGP nor FS-BGP can guarantee route authenticity for networks in which there is at least a single node without a public key, and then present variants of these protocols with which full security can be guaranteed in partial PKI scenarios.

8.1 Partial PKI Deployment: Introductory Results

Before stating our main introductory result in Theorem 8.4, to develop intuition as to why providing security guarantees in scenarios with partial PKI deployment is a very difficult problem, we present a simple example of an attack where only a one AS has no public key and only one AS is corrupted. First, we formalize the modification of allowing some nodes not to have public keys in S-BGP as follows.



Fake Link Attack

Figure 4: N_1 does not have a public key, and the adversary corrupts only N_3; in this route authentication attack N_3 takes N_2 out of the route and announces a shorter, infeasible route to N_4.

Construction 8.1. *Let* $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$ *be an interdomain network and* k *a security parameter. We define S-BGP with partial deployment (S-BGP-PD)* $= (\mathsf{Init}', \mathsf{An}')$ *as a path vector protocol identical to S-BGP* $= (\mathsf{Init}, \mathsf{An})$ *but with the following modifications. During execution of* $\mathsf{Init}(1^k)$ *not every node has to generate a public key. During execution of* An'*, nodes that do not have public keys do not generate route attestations, and route attestations of nodes without public keys are not checked during the route attestation verification.*

Notice that S-BGP is just a special case of S-BGP-PD when all nodes have keys. In Figure 4 we present a simple route authentication attack that shows that S-BGP-PD does not guarantee route authentication when $m = 1$, for $\mathcal{C}_1^{\text{S-BGP-PD}}$. The attack consists of the adversary taking an intermediate node N_2 out of the route announcement during the execution of S-BGP-PD. This results in N_4 accepting a shorter, infeasible route, so in this scenario $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\text{S-BGP-PD}}(A)$ returns 2 with probability 1.

Remark 8.2. *Given an attack in a network with* m *nodes without public keys one can always construct an attack in a network with* m' *nodes without public keys, for any* $m' > m$*, by making more nodes keyless in the same network.*

This is because increasing the number of keyless nodes cannot make a feasible attack infeasible. Without affecting the attack, the number of nodes with public keys in the network can be increased by adding neighbors to an origin. Thus, the attack in Figure 4 shows that for no $m \geq 1$ does S-BGP-PD guarantee route authentication with m-PD for $\mathcal{C}_m^{\text{FS-BGP-PD}}$.

Providing security guarantees in scenarios with partial PKI deployment is a difficult problem because nodes that do not have public keys cannot generate route attestations. The attack in Figure 4 works because S-BGP-PD does not guarantee route feasibility since nodes cannot find out using this protocol whether some remote nodes are neighbors. When not all nodes have public keys, providing nodes with capabilities of verifying neighborship of remote nodes ultimately would require a certificate from a third trusted party, such as link attestations in FS-BGP. Let us define FS-BGP-PD to account for partial PKI deployment similarly to Construction 8.1. Notice that FS-BGP is just a special case of FS-BGP-PD when all nodes have keys. It can be easily shown that the route authentication attack in Figure 4 would not be possible if nodes used FS-BGP-PD to establish a route to P.

Remark 8.3. *FS-BGP-PD guarantees origin authentication and route validity with* m*-PD for any network in* $\mathcal{C}_m^{\mathit{FS\text{-}BGP\text{-}PD}}$*, for any* $m \leq |\mathsf{AS's}|$*, if the underlying* $\mathcal{CP}$ *is uf-cda-secure.*

This is because for networks in $\mathcal{C}_m^{\text{FS-BGP-PD}}$, in FS-BGP-PD origin authentication and route validity do not depend on whether nodes have public keys or not. However, the following result shows that even when origin authentication and route validity are guaranteed, route authentication cannot be guaranteed when $|\mathsf{nopubk}| > 0$.

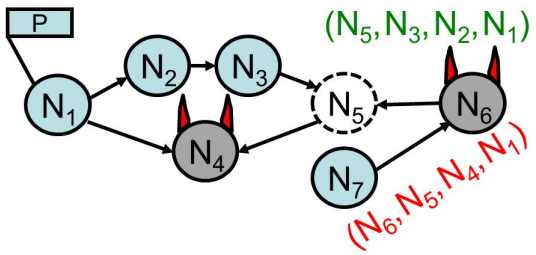


Figure 5: Only N_5 does not have a public key, and the adversary corrupts N_4 and N_6. In this VRS route authentication attack N_6 announces to N_7 a valid route to P that N_5 did not authorize N_6 to announce.

Theorem 8.4. *For no $m \geq 1$ does FS-BGP-PD guarantee route authentication with m-PD for $\mathcal{C}_m^{\text{FS-BGP-PD}}$.*

Proof. Consider a network which contains at least one node N_i which does not have a public key, has a choice of at least two routes to the same prefix, and has a neighbor N_j whose only access to that prefix is through N_i and to whom N_i is willing to export at least two different routes to that prefix.

We construct an efficient adversary A such that $\mathbf{Adv}^{\text{sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PD}}(A) = 1$ as follows. Since A can observe all communication, it can learn of all the routes available to N_i as well as the route N_i announces to N_j. A can intercept N_i's announcement to N_j and switch the route in that announcement to another valid route, available to N_i, that N_i is willing to export to N_j. Since N_i does not have a public key, it cannot generate a route attestation for its original announcement, so N_j is bound to accept this false announcement that contains a valid route. Therefore, $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\text{S-BGP-PD}}(A)$ returns 2 with probability 1 in this scenario, so $\mathbf{Adv}^{\text{sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PD}}(A) = 1$. The theorem then follows due to Remark 8.2. □

To show a pictorial example of the proof of Theorem 8.4, in Figure 5 we present an attack where the adversary switches a valid route announced by node N_5 without a public key for another valid route that N_5 never announced. The adversary corrupts two nodes, N_4 and N_6. In this network, N_5 prefers the longer customer route through N_3 to the provider route through N_4 (recall that a directed edge from one node to another indicates that the former pays the latter for all traffic exchanged on their link). However, N_6 switches N_5's more preferred route to the one through N_4 in its announcement to N_7, who accepts this route as authentic since N_5 does not have a public key (and thus cannot generate a route attestation). Therefore, $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\text{S-BGP-PD}}(A)$ returns 2 with probability 1 in this scenario. By Remark 8.2 the same attack can be carried out for any $m > 1$.

The attack in the proof of Theorem 8.4, deserves a special name because we later show it to be the only type of attacks that can prevent FS-BGP-PD from being fully secure later in this section. A similar type of attack was known in the networking community to prevent SoBGP from guaranteeing route authentication.

Definition (The Valid-Route-Switching Attack)**.** *Let $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$ be a network in $\mathcal{C}_m^{\mathcal{PV}}$, for any $1 \leq m \leq |\mathsf{AS's}|$, such that $|\mathsf{AS's}| \geq 2$, let $\mathcal{PV} = (\mathsf{Init}, \mathsf{An})$ be a path vector protocol correct for $\mathcal{I}$ and let k be the security parameter such that*

the size of the description of $\mathcal{I}$ is polynomial in k. We consider the experiment $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$, involving an adversary A.

When $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ outputs 2, i.e. when $N_\ell \in$ Honest accepts announcement $(N_{\ell-1}, N_\ell, R = (N_{\ell-1}, \ldots, N_1), P, W, Aux)$, such that $\exists 1 \leq i \leq \ell - 1$ so that $N_i \in$ Honest has never output announcement $(N_i, N_{i+1}, R' = (N_i, \ldots, N_1), P, W', Aux')$ for any W', Aux' to N_{i+1}, if in addition $N_i \in$ nopubk and R' is a valid route to P, then this event is called a Valid-Route-Switching (VRS) attack.

In the definition of the Valid-Route-Switching (VRS) attack, an honest node N_i may never announce to N_{i+1} a valid route R' to a particular prefix P because N_i may have never received any route announcements to P from its neighbors or because R' is not N_i's most preferred route to P. Notice that VRS attacks can cause subroute inconsistency, so they can cause FS-BGP-PD to come to a stable but inconsistent state as in the example in the proof of 8.4, where FS-BGP-PD does not diverge since all nodes select their most preferred routes after a finite number of transmitted route announcements, but there is an inconsistency between the preferred routes of N_7 and N_5.

8.2 The Relaxed Path Vector Protocol Security Definition

We first motivate two relaxations to our security definition. We justify that these relaxations are in fact reasonable on the Internet of today due to physical security of communication links and the trust relationship that neighboring AS's can establish when they agree to form business relationships. We then formalize and integrate these restrictions into our security model, in a form of restrictions on the adversary, to present a new security definition for path vector protocols adequate for scenarios with partial PKI deployment. Finally, we present refinements to S-BGP-PD and FS-BGP-PD that address the weakness pointed out in the proof of Theorem 8.4, and prove that the refined protocols meet our new definition.

Currently available technology allows honest neighboring AS's, whether with public keys or not, to establish communication channels that guarantee authentication and integrity. AS's could establish communication channels with their neighbors via IPSec that could guarantee integrity and authenticity, for which they do not need public keys as they could establish pre-shared keys off line, since they would have to establish a business relationship to have a physical connection anyway. BGP TTL security hack [53] could also be used for this purpose. Although most of the time AS's establish connections at Internet Exchange Points (IXP), sometimes connections between AS's are established via fiber-optic cables outside of IXP's. Such cables mostly run underground and may be closely monitored for performance deviations. The transmitted data along such cables is transformed into optical signals that are impossible to interpret without expensive equipment. Thus, although attacking such cables is feasible in principle, it would require an impractical amount of resources in real life. Sender authenticity could be added with appropriate gateway configurations, which associate neighboring AS's to specific outgoing and incoming ports, such that announcements get dropped when they come to the port not associated with the neighbor claiming to have sent them.

Note that, although other types of physical attacks on links between nodes are possible and have been studied before [16], these types of attacks do not involve listening and intercepting data without being noticed. The only purpose of these attacks is to take out links out of a topology so that certain route announcements are never made.

To establish a business relationship between themselves, neighboring AS's must be able to establish some level of trust between each other. Many AS's on the Internet are now multi-homed,

so framing AS business partners on the Internet could lead to unwanted consequences such as the tearing down of their business contracts and physical links connecting them, which could result in substantial financial losses. Having established trust with their neighbors, AS's that do not have public keys could rely on their trusted down-stream neighbors with public keys to "vouch" for the former with their signatures.

As mentioned above, on the Internet, most connections between AS's are made at public or private IXP which, intuitively, serve the role of rendez-vous points for AS's to exchange traffic. AS's that wish to connect at a particular IXP have to establish a physical connection at that IXP. Thus, since IXP's make a profit by providing basic infrastructure for AS's to make connections and become neighbors, it would be in their interest to facilitate the establishment of physically secure communication channels and trust between neighboring AS's, as this would guarantee longer lasting business relationships for those AS's (which would imply longer lasting profits for the IXP connecting them).

We formally present these two main points in the following two relaxations.

Security Relaxations.

1. (Physical-Link-Security Relaxation) A is not allowed to (i) send announcements on behalf of honest neighboring nodes and (ii) intercept and modify announcements exchanged between neighboring honest nodes.

2. (Trusted-Next-Neighbor Relaxation) Whenever experiment $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ outputs 2, i.e. $N_\ell \in \mathsf{Honest}$ accepts announcement $(N_{\ell-1}, N_\ell,\ R = (N_{\ell-1}, \ldots, N_1),\ P, W, Aux)$, and there exists $1 \leq i \leq \ell - 1$ such that $N_i \in \mathsf{Honest}$ never output $(N_i,\ N_{i+1}, R' = (N_i, \ldots, N_1), P,\ W', Aux')$ for any W', Aux' to N_{i+1}, $N_{i+1} \in \mathsf{Honest}$ if $N_i \in \mathsf{nopubk}$.

In what follows, we incorporate Relaxations 1-2 into a new security definition for path vector protocols where adversary's behavior is restricted according to these relaxations.

The Relaxed Security Definition. We relax the security definition from Section 5 as follows.

Definition. *Let* $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}),\ \mathsf{Prefixes},\ \mathsf{OrforPr},\ \mathsf{relation},\ \mathsf{preferto},\ \mathsf{policy})$ *be a network in* $\mathcal{C}_m^{\mathcal{PV}}$*, for any* $1 \leq m \leq |\mathsf{AS's}|$*, such that* $|\mathsf{AS's}| \geq 2$*, let* $\mathcal{PV} = (\mathsf{Init}, \mathsf{An})$ *be a path vector protocol and let* k *be the security parameter such that the size of the description of* $\mathcal{I}$ *is polynomial in* k*. We define experiment* $\mathbf{Exp}^{\text{r-sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ *involving adversary* A *to be identical to the experiment* $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ *involving an adversary* A *from the definition from Section 5 except that Relaxations 1-2 must hold.*

We define A's advantage $\mathbf{Adv}^{\text{r-sec-rout-m-b}}_{\mathcal{I},\mathcal{PV}}(A)$ in this experiment as $\Pr\left[\mathbf{Exp}^{\text{r-sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A) = b\right]$, for $b \in \{1, 2, 3\}$. We say that $\mathcal{PV}$ guarantees *relaxed origin authentication*, *route authentication*, and *route validity* with m-PD for a class of networks $\mathcal{C}_m^{\mathcal{PV}}$, if for every network $\mathcal{I} \in \mathcal{C}_m^{\mathcal{PV}}$, for every efficient adversary A the probability experiment $\mathbf{Exp}^{\text{r-sec-rout-m}}_{\mathcal{I},\mathcal{PV}}(A)$ returns 1, 2 and 3 respectively, while Relaxations 1-2 hold, is negligible in k. The *relaxed full security* is defined analogously to security definition in Section 5.

Secure Constructions. We slightly modify S-BGP-PD and then show that it meets the above definition.

Construction 8.5. *Let* $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$ *be an interdomain network. We define S-BGP-PD with a restriction (S-BGP-PDR)* $= (\mathsf{Init}, \mathsf{An}')$ *as a path vector protocol identical to S-BGP-PD* $= (\mathsf{Init}, \mathsf{An})$ *but with the following restrictions in* An'*. When a node receives an announcement of a route, that node rejects the announcement if that route contains more than one node without public keys in a row at any part of that route. Also, a node without a public key does not propagate a route that was announced by its neighbor who also does not have a public key.*

We define FS-BGP-PD with a restriction (FS-BGP-PDR) similarly; note that in S-BGP-PDR and FS-BGP-PDR, the last two nodes on a route could be without public keys. This new restriction implicitly requires that nodes reject announcements that are missing a signature for at least one node in that route who has a public key. Although checking whether a node has a public key or not may be difficult in practice, this is in fact necessary, otherwise an adversarial node could simply strip an honest node's signature and send a bogus route on its behalf.

Theorem 8.6. *S-BGP-PDR as defined in Construction 8.5 guarantees relaxed route authentication with m-PD for* $\mathcal{C}_m^{\textit{S-BGP-PDR}}$*, for any* $m \leq |\mathsf{AS's}|$*, if the underlying* $\mathcal{SS}$ *is uf-cma-secure.*

Proof. We show that for every adversary A attacking route authentication of S-BGP-PDR, there exist adversaries B and C attacking unforgeability of $\mathcal{SS}$ such that

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS}}(B) + \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS}}(C) \geq \frac{1}{|\mathsf{AS's}|}\mathbf{Adv}^{\text{r-sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PDR}}(A),$$

and the resources of each are that of A plus overhead upper bounded by the size network using S-BGP-PDR that A is attacking.

Suppose $\mathcal{C}_m^{\text{S-BGP-PDR}} \neq \emptyset$ and let A be an efficient adversary attacking route authentication of S-BGP-PDR for a network $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy}) \in \mathcal{C}_m^{\text{S-BGP-PDR}}$, $1 \leq m \leq |\mathsf{AS's}|$ and $|\mathsf{AS's}| \geq 2$, whose description is polynomial in k.

As a result of A's attack, $N_\ell \in \mathsf{Honest}$ accepts an announcement $(N_{\ell-1}, N_\ell, R = (N_{\ell-1}, \ldots, N_1), P, W, Aux)$, such that $\exists\ 1 \leq i \leq \ell - 1$ so that $N_i \in \mathsf{Honest}$, N_i has never sent announcement $(N_i, N_{i+1}, R' = (N_i, N_{i-1}, \ldots, N_1), P,\ W', Aux')$, for some W', Aux', and $N_{i+1} \in \mathsf{Honest}$ if $N_i \in \mathsf{nopubk}$, while Relaxations 1-2 hold. We refer to this event by A frames i.

Notice that either N_i has a public key or it does not. $N_\ell \in \mathsf{Honest}$ could not have accepted a route announcement with two nodes without a public key in a row, so, by construction of S-BGP-PDR and Relaxation 2, N_{i+1} must have a public key and be honest if $N_i \in \mathsf{nopubk}$ and $i < \ell - 1$. Since $N_i \in \mathsf{Honest}$, N_i would not send an announcement to a node that is not its neighbor, so there must be a link between N_i and N_{i+1}. If N_{i+1} has never accepted the announcement that N_i has never actually sent, it must be that $i < \ell - 1$, since, by definition of the attack, N_ℓ did accept $(N_{\ell-1}, N_\ell, R = (N_{\ell-1}, \ldots, N_1), P, W, Aux)$. If N_{i+1} did accept it (which means that it must have received it), A must have either generated that announcement and sent it on behalf of N_i or intercepted and modified some other N_i's announcement. However, this cannot happen as it would violate Relaxation 1, in which case A would not win (note that this also includes the case when $i = \ell - 1$). More concretely, exactly one of the following two conditions must hold when A frames i:

(1) $N_i \notin \mathsf{nopubk}$ or

(2) $N_i \in \mathsf{nopubk}$, $i < \ell - 1$, and N_{i+1} never accepted announcement $(N_i, N_{i+1}, R' = (N_i, N_{i-1}, \ldots, N_1), P,\ W', Aux')$, for any W', Aux'.

When condition (1) holds, we construct adversary B attacking unforgeability of $\mathcal{SS}$ as follows. B is given a public key pk and the signing oracle $\mathsf{Sign}(K,\cdot)$ in $\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B)$. After giving A the description of $\mathcal{I}$, who then selects the set $\mathsf{nopubk} \subsetneq \mathsf{AS's}$ of nodes who will not have public keys, B picks a node at random $N_x \xleftarrow{\$} \mathsf{AS's}$ and then generates public-private key pairs for all nodes not in $\mathsf{nopubk} \cup \{N_x\}$ using $\mathsf{Kg}(1^k)$. B then sets $\mathbf{pk}[x] \leftarrow pk$ and gives A all the public keys. A outputs initial partition $(\mathsf{Honest}, \mathsf{Corrupted})$ of $\mathbf{G}$. If $N_x \in \mathsf{Corrupted}$, then B aborts its attack. Otherwise B gives A all the secret keys of the corrupted nodes. The rest of the proof for this condition is identical to that of Theorem 6.3. Therefore, $\Pr\left[\,\text{cond 1} \;\middle|\; \mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B) = 1\,\right]$ that B wins by outputting $((N_{i+1}, N_i, \ldots, N_1, P), RA_{R_i^{i+1}})$, when condition (1) holds, over all $i \in \mathsf{AS's}$, is $\frac{1}{|\mathsf{AS's}|}\sum_{i\in\mathsf{AS's}} \Pr[\,\text{cond 1} \mid A \text{ frames } i\,]$. B is efficient since, to simulate S-BGP-PDR, the number of queries it makes to $\mathsf{Sign}(K,\cdot)$ and their length are upper-bounded by the number of queries that A makes and the size of $\mathcal{I}$ respectively.

When condition (2) is true, we construct adversary C attacking unforgeability of $\mathcal{SS}$ the same way as adversary B when condition (1) is true, only in this case, at the end of A's attack, C would output $((N_{i+2}, N_{i+1}, \ldots, N_1, P), RA_{R_{i+1}})$. Note that since N_{i+1} never accepted $(N_i, N_{i+1}, R' = (N_i, N_{i-1}, \ldots, N_1), P, W', Aux')$, N_{i+1} could not have sent $(N_{i+1}, N_{i+2}, (N_{i+1}, R'), P, W'', Aux'')$ to N_{i+2} because $N_{i+1} \in \mathsf{Honest}$ due to Relaxation 2. Therefore, C's output is "new" in the sense that C never queried $((N_{i+2}, N_{i+1}, \ldots, N_1, P)$ to the signing oracle. Thus, $\Pr\left[\,\text{cond 2} \;\middle|\; \mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(C) = 1\,\right]$ is also $\frac{1}{|\mathsf{AS's}|}\sum_{i\in\mathsf{AS's}} \Pr[\,\text{cond 2} \mid A \text{ frames } i\,]$.

C is efficient since, to simulate S-BGP-PDR, the number of queries it makes to $\mathsf{Sign}(K,\cdot)$ and their length are upper-bounded by the number of queries that A makes and the size of $\mathcal{I}$ respectively.

We thus have that

$$
\begin{aligned}
\mathbf{Adv}^{\text{r-sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PDR}}(A) &= \Pr\left[\,\mathbf{Exp}^{\text{r-sec-rout-m}}_{\mathcal{I},\text{S-BGP-PDR}}(A) = 2\,\right] \\
&= \sum_{j=1}^{2}\left(\sum_{i\in\mathsf{AS's}} \Pr[\,\text{cond j} \mid A \text{ frames } i\,]\Pr[\,\text{cond j}\,]\right) \\
&= |\mathsf{AS's}|\Pr\left[\,\text{cond 1} \;\middle|\; \mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B) = 1\,\right]\Pr[\,\text{cond 1}\,] \\
&+ |\mathsf{AS's}|\Pr\left[\,\text{cond 2} \;\middle|\; \mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(C) = 1\,\right]\Pr[\,\text{cond 2}\,] \\
&\le \sum_{j=1}^{2} |\mathsf{AS's}|\Pr\left[\,\text{cond j} \;\middle|\; \mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B) = 1\,\right]\Pr[\,\text{cond j}\,] \\
&+ \sum_{j=1}^{2} |\mathsf{AS's}|\Pr\left[\,\text{cond j} \;\middle|\; \mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(C) = 1\,\right]\Pr[\,\text{cond j}\,] \\
&= |\mathsf{AS's}|\left(\Pr\left[\,\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B) = 1\,\right] + \Pr\left[\,\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(C) = 1\,\right]\right)\,.
\end{aligned}
$$

□

Corollary 8.7. *FS-BGP-PDR is relaxed fully secure with m-PD for $\mathcal{C}_m^{FS\text{-}BGP\text{-}PDR}$, for $m \le |\mathsf{AS's}|$, if the underlying $\mathcal{SS}$ and $\mathcal{CP}$ are uf-cma-secure and uf-cda-secure respectively.*

Proof. The proof follows from Theorems 7.2 and 8.6 and Remark 8.3. □

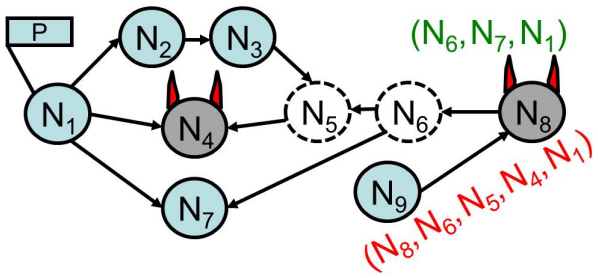


Figure 6: N_5 and N_6 do not have public keys, and the adversary corrupts N_4 and N_8. In this VRS route authentication attack N_8 announces to N_9 a valid route to P that N_5 never authorized N_6 to announce. Note that Relaxations 1-2 are satisfied since N_6 is honest and the adversary does not need to intercept and modify communication between honest nodes.

A significant practical implication of Theorem 8.6 and Corollary 8.7 is that new AS's who have just joined the Internet but do not have public keys, do not have to get a public key as long as they establish a trust relationship with their neighbors in the sense that for any route announcement that they make, they are sure that their neighbors who have public keys will vouch for them.

The following results emphasize that the restrictions in the relaxed path vector protocol security definition posed by Relaxations 1-2 and the requirement to ignore routes that have more than one node without a public key in a row, as is done in S-BGP-PDR and FS-BGP-PDR, are in fact necessary. The latter restriction, in the worst case, could cause some parts of the network to become disconnected as many routes may be ignored.

Theorem 8.8. *For the statements in Theorem 8.6 and Corollary 8.7 to hold, each relaxation (Physical-Link-Security or Trusted-Next-Neighbor) is necessary given the other one.*

Proof. The proof is demonstrated in Figure 5. If the Trusted-Next-Neighbor relaxation does not hold, then the adversary can perform the same attack as in the proof of Theorem 8.4. If the Physical-Link-Security relaxation does not hold, then the adversary can do the same by intercepting and modifying the route announcement on a link between N_5 and N_6, while corrupting no node. In either case, $\mathbf{Adv}^{\text{sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PDR}}(A) = 1$ and A is efficient. □

Theorem 8.9. *Even when the underlying $\mathcal{SS}$ is uf-cma-secure, S-BGP-PD as per Construction 8.1 and FS-BGP-PD do not guarantee relaxed route authentication with m-PD for $\mathcal{C}_m^{\text{S-BGP-PDR}}$ and $\mathcal{C}_m^{\text{FS-BGP-PDR}}$ respectively, for any $m \geq 2$.*

Proof. This is shown in Figure 6, for $m = 2$. While N_5 prefers a customer route through N_3, N_6 prefers a shorter route through N_7. On behalf of N_8, the adversary announces a valid route to N_9 that goes through N_4. N_9 accepts this route, because there is no way for N_9 to find out that N_5 never announced to N_6 a route through N_4. This is because neither N_5 nor N_6 has a public key, although both are honest. Observe that in this case $\mathbf{Adv}^{\text{r-sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PD}}(A) = 1$ and A is efficient. □

Even if we do not rely on security Relaxation 2, we can still show that it is possible to guarantee route authentication but with a very restricted version of S-BGP-PD, where only the last two nodes on any route are allowed not to have public keys. We provide the details in Appendix B.

8.3 What if there is no PKI

We show that if all prefixes and links are certified by a trusted certification authority, even when no node has a public key, nodes are guaranteed to discover valid routes with authentic origins, and that VRS attacks are the *only* attacks that prevent FS-BGP-PD from guaranteeing route authentication. In light of this result, we then discuss the feasibility of achieving reasonable security without PKI.

Theorem 8.10. *If the underlying $\mathcal{SS}$ is uf-cma-secure and the underlying $\mathcal{CP}$ is uf-cda-secure, for any $1 \leq m \leq |\mathsf{AS's}|$, if $\mathbf{Exp}^{\text{sec-rout-m}}_{\mathcal{I},FS\text{-}BGP\text{-}PD}(A) = 2$ (see security definition in Section 5), then A must have carried out a VRS attack.*

Proof. Let us define advantage $\mathbf{Adv}^{\text{r-sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PD,no-VRS}}(A)$ of any adversary A attacking route authentication of FS-BGP-PD to be the probability $\Pr\left[\text{no VRS} \;\middle|\; \mathbf{Exp}^{\text{r-sec-rout-m}}_{\mathcal{I},\text{FS-BGP-PD}}(A) = 2\right]$ that A wins without performing a VRS attack. We show that for every adversary A attacking route authentication of FS-BGP-PD, there exist adversaries B and C attacking unforgeability of $\mathcal{SS}$ and $\mathcal{CP}$ respectively such that $\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS}}(B) + \frac{1}{|\mathsf{AS's}|}\mathbf{Adv}^{\text{uf-cda}}_{\mathcal{CP}}(C) \geq \frac{1}{|\mathsf{AS's}|}\mathbf{Adv}^{\text{r-sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PD,no-VRS}}(A)$ and the resources of each are at most that of A plus some overhead upper bounded by the size network using FS-BGP-PD that A is attacking.

Suppose $\mathcal{C}^{\text{FS-BGP-PD}}_m \neq \emptyset$ and let A be an efficient adversary attacking route authentication of FS-BGP-PD for a network $\mathcal{I} \in \mathcal{C}^{\text{FS-BGP-PD}}_m$ with $m \geq 1$ and $|\mathsf{AS's}| \geq 2$, whose description is polynomial in k.

As a result of A's attack, $N_\ell \in \mathsf{Honest}$ accepts an announcement $(N_{\ell-1}, N_\ell, R = (N_{\ell-1}, \ldots, N_1), P, W, Aux)$, such that $\exists\, 1 \leq i \leq \ell-1$ and $N_i \in \mathsf{Honest}$ has never sent announcement $(N_i, N_{i+1}, R' = (N_i, N_{i-1}, \ldots, N_1), P,\ W', Aux')$, for some W', Aux', to N_{i+1}. Without loss of generality, let us consider the closest such N_i to the origin N_1. The following are all the possible reasons for why N_i would not send that announcement to N_{i+1}.

1. $(N_{i+1}, N_i, \ldots, N_1)$ is a valid route to P, but N_i has never received $(N_{i-1}, N_i, (N_{i-1}, N_{i-2}, \ldots, N_1), P, \ldots)$;
2. N_i has received $(N_{i-1}, N_i, (N_{i-1}, N_{i-2}, \ldots,\ N_1), P, \ldots)$ but rejected it;
3. N_i has received and accepted $(N_{i-1}, N_i, (N_{i-1}, N_{i-2}, \ldots,\ N_1), P, \ldots)$, but N_i did not announce R' to N_{i+1} because
   (a) $N_{i+1} \notin \mathsf{Neighbors}(N_i)$,
   (b) $N_{i+1} \notin \mathsf{policy}_{N_i}((N_i, N_{i-1}, N_{i-2}, \ldots, N_1), \mathsf{relation}(N_i, N_{i-1}))$,
   (c) $(N_{i-1}, \ldots, N_1)$ is not N_i's preferred route to P.

If N_i has a public key, then N_ℓ would accept $(N_{\ell-1},\ N_\ell,\ R = (N_{\ell-1}, \ldots, N_1),\ P, W, Aux)$ only after checking the validity of N_i's route attestation of R', so in this case we could construct an adversary B attacking the unforgeability of the underlying $\mathcal{SS}$ the same way as in proof of condition (1) of Theorem 8.6. B is given a public key pk and the signing oracle $\mathsf{Sign}(K, \cdot)$ in $\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B)$. After giving A the description of $\mathcal{I}$, who then selects the set $\mathsf{nopubk} \subsetneq \mathsf{AS's}$ of nodes without public keys, B picks a node at random $N_x \xleftarrow{\$} \mathsf{AS's}$ and then generates public-private key pairs for all nodes not in $\mathsf{nopubk} \cup \{N_x\}$ using $\mathsf{Kg}(1^k)$. B then sets $\mathbf{pk}[x] \leftarrow pk$ and gives A all the public keys. A outputs initial partition $(\mathsf{Honest}, \mathsf{Corrupted})$ of $\mathbf{G}$. If $N_x \in \mathsf{Corrupted}$, then B aborts its attack.

Otherwise B gives A all the secret keys of the corrupted nodes. The rest of the proof for this condition is identical to that of Theorem 6.3. Therefore, $\Pr\left[N_i \notin \mathsf{nopubk} \;\middle|\; \mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B) = 1 \right]$ that B wins by outputting $((N_{i+1}, N_i, \ldots, N_1, P),\ RA_{R_i^{i+1}})$ when N_i has a public key, over all $i \in \mathsf{AS's}$, is $\frac{1}{|\mathsf{AS's}|} \sum_{i \in \mathsf{AS's}} \Pr[\, N_i \notin \mathsf{nopubk} \mid A \text{ frames } i \,]$. B is efficient since, to simulate FS-BGP-PD, the number of queries it makes to $\mathsf{Sign}(K, \cdot)$ and their length are upper-bounded by the number of queries that A makes and the size of $\mathcal{I}$ respectively.

Because FS-BGP-PD guarantees origin authentication and route validity (see Theorem 7.2 and Remark 8.3), if $N_i \in \mathsf{nopubk}$, then A must succeed in a VRS attack only. This is because if reason 2 holds, then either $(N_{i-1}, N_{i-2}, \ldots, N_1)$ is invalid or $\mathsf{OrforPr}(P) \neq N_1$ (recall that we have chosen N_i to be the closest framed node to the origin). Note that reason 2 also contains less interesting issues such as lack of a route attestation from some intermediate node $N_j \notin \mathsf{nopubk}$, for $1 \leq j < i$, in the announcement or a bogus route/address attestation that does not verify during S-BGP's attestation verification steps (see Construction 4.1). However, if N_i would not accept this announcement due to any of these issues, then neither would N_ℓ, since both N_i and N_ℓ are honest. If either of the reasons 3(a) or 3(b) holds, then (N_{i+1}, R') is invalid. R must be invalid if at least one of its subroutes, in this case (N_{i+1}, R'), is invalid. Thus, if any one of reasons 2, 3(a) or 3(b) is true, since $N_\ell \in \mathsf{Honest}$, it could not have accepted announcement $(N_{\ell-1},\ N_\ell,\ R = (N_{\ell-1}, \ldots, N_1),\ P, W, Aux)$, since FS-BGP-PD guarantees origin authentication and route validity. If $N_i \in \mathsf{nopubk}$ and only one of reasons 1 or 3(c) is true, then (N_{i+1}, R') is a valid route to P and A succeeds in a VRS attack. (Note that reason 3(c) contains the scenario in which R contains N_i, in which case N_i would ignore R as a loop-preventative measure.)

Thus, if $N_i \in \mathsf{nopubk}$ and A does not succeed in a VRS attack, then we can construct adversary C attacking the unforgeability of the underlying $\mathcal{CP}$ as follows. C is given CA's public key pk_{CA} in $\mathbf{Exp}^{\text{uf-cda}}_{\mathcal{CP}}(B)$. C first gives A the description of $\mathcal{I}$, who then selects the set $\mathsf{nopubk} \subseteq \mathsf{AS's}$ of nodes who will not have public keys. C then generates public-private key pairs for all nodes not in nopubk using $\mathsf{Kg}(1^k)$. C then gives A all the public keys, including that of the CA. A outputs initial partition $(\mathsf{Honest}, \mathsf{Corrupted})$ of $\mathbf{G}$. C gives A all the secret keys of the corrupted nodes. A starts the execution of FS-BGP-PD on behalf of all nodes in $\mathsf{Corrupted}$ together with C who executes FS-BGP-PD on behalf of all nodes in Honest and the CA. C follows FS-BGP-PD legitimately, whereas A is allowed to act arbitrarily. C stores all the communication and provides it to A.

For each node N_i and prefix P, such that N_i owns P (C can check this with $\mathsf{OrforPr}$), where either $N_i \in \mathsf{Honest}$ or $N_i \in \mathsf{Corrupted}$ and A has requested address attestation $AA^P_{N_i}$ of P on behalf of N_i, C sequentially interacts with the CA via $(\mathsf{CA}(K_{\mathsf{CA}}, N_i, P),\ B(pk_{\mathsf{CA}},\ N_i,\ P))$ to get (N_i, P, AA^P_i). Similarly, for each node N_i and its neighbor N_j (C can check this with link), where either $N_i \in \mathsf{Honest}$ or $N_i \in \mathsf{Corrupted}$ and A has requested link attestation $LA^P_{N_iN_j}$ on behalf of N_i, C sequentially interacts with the CA via $(\mathsf{CA}(K_{\mathsf{CA}}, N_i, ((\min(N_i, N_j),\ \max(N_i, N_j)),\ \mathsf{relation}(\min(N_i, N_j), \max(N_i, N_j)))),\ B(pk_{\mathsf{CA}}, N_i, ((\min(N_i, N_j), \max(N_i, N_j)), \mathsf{relation}(\min(N_i, N_j),\ \max(N_i, N_j)))))$ to get $(N_j, ((\min(N_j, N_i), \max(N_j, N_i)), \mathsf{relation}(\min(N_j, N_i), \max(N_j, N_i))), LA_{N_iN_j})$. C stores all address and link attestations. This information together with all honest nodes' secret keys, allows C to follow the computations according to the interactive algorithm An. Observe that C's simulation for A is perfect.

When A outputs $(N_{\ell-1},\ N_\ell,\ R = (N_{\ell-1}, \ldots, N_1),\ P, W, Aux)$, such that $N_\ell \in \mathsf{Honest}$ accepts it and $\exists\ 1 \leq i \leq \ell - 1$ an $N_i \in \mathsf{Honest}$ has never sent announcement $(N_i, N_{i+1}, R' = (N_i, N_{i-1}, \ldots, N_1), P,\ W', Aux')$, for some W', Aux', C proceeds as follows. If R' is invalid, then there must be $1 < j < i$ such that either $\mathsf{link}(N_j, N_{j+1}) = 0$, in which case C outputs $(N_j,\ (N_j, N_{j+1},$

rel, $LA_{N_jN_{j+1}}$), where rel is a fake relationship between N_j and N_{j+1} since they are not neighbors but is presented in Aux', or $\mathsf{link}(N_j, N_{j+1}) = 1$ but $N_{j+1} \notin \mathsf{policy}_{N_j}((N_j, N_{j-1}, \ldots, N_1), \mathsf{relation}(N_j, N_{j-1}))$ in which case C outputs $(N_{j-1}, (N_{j-1}, N_j, \text{rel}, LA_{N_{j-1}N_j}))$, where rel $\neq \mathsf{relation}(N_{j-1}, N_j)$ but is presented in Aux'. Note that these are the only reasons why R' would not be valid, since policy is publicly available in our model, so if N_i would not accept the announcement with R', then neither would N_ℓ accept the announcement with R for the same reason. Otherwise, if N_1 does not own P, then C outputs (N_1, P, AA_1^P), where AA_1^P must be the last entry of Aux'. Otherwise, if reason 3(a) is true, then C outputs $(N_i, (N_i, N_{i+1}, \text{rel}, LA_{N_iN_{i+1}})$, whee rel is a fake relationship between N_i and N_{i+1} but is presented in Aux'. Otherwise, if reason 3(b) is true, then C outputs $(N_{i-1}, (N_{i-1}, N_i, \text{rel}, LA_{N_{i-1}N_i}))$ where rel $\neq \mathsf{relation}(N_{i-1}, N_i)$ but is presented in Aux'.

Since when $N_i \in \mathsf{nopubk}$ reasons 2-3(b) above cover all possible non-VRS-attack events that could occur, C's probability of breaking uf-cda security of $\mathcal{CP}$ is the same as that of A breaking route authenticity of $\mathcal{PV}$:
$\Pr\left[\mathbf{Exp}^{\text{uf-cda}}_{\mathcal{CP}}(C) = 1\right] = \sum_{i \in \mathsf{AS's}} \Pr\left[N_i \in \mathsf{nopubk} \mid A \text{ frames } i\right]$.
Note that C is as efficient as A.

Thus we have that if A does not succeed in a VRS attack, then

$$\begin{aligned}
\mathbf{Adv}^{\text{r-sec-rout-m-2}}_{\mathcal{I},\text{S-BGP-PD,no-VRS}}(A) &= \sum_{i \in \mathsf{AS's}} \Pr\left[N_i \notin \mathsf{nopubk},\ \text{no VRS} \mid A \text{ frames } i\right] \Pr\left[N_i \notin \mathsf{nopubk},\ \text{no VRS}\right] \\
&+ \sum_{i \in \mathsf{AS's}} \Pr\left[N_i \in \mathsf{nopubk},\ \text{no VRS} \mid A \text{ frames } i\right] \Pr\left[N_i \in \mathsf{nopubk},\ \text{no VRS}\right] \\
&= |\mathsf{AS's}| \Pr\left[N_i \notin \mathsf{nopubk},\ \text{no VRS} \;\middle|\; \mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B) = 1\right] \Pr\left[N_i \notin \mathsf{nopubk},\ \text{no VRS}\right] \\
&+ \Pr\left[N_i \in \mathsf{nopubk},\ \text{no VRS} \;\middle|\; \mathbf{Exp}^{\text{uf-cda}}_{\mathcal{CP}}(C) = 1\right] \Pr\left[N_i \in \mathsf{nopubk},\ \text{no VRS}\right] \\
&\leq |\mathsf{AS's}| \Pr\left[\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{SS}}(B) = 1\right] + \Pr\left[\mathbf{Exp}^{\text{uf-cda}}_{\mathcal{CP}}(C) = 1\right].
\end{aligned}$$

□

The goal of path vector protocols is for AS's to learn of routes in the network to all prefixes, so the importance of Theorem 8.10 is that FS-BGP-PD guarantees that AS's learn of valid routes with authentic origins and that, even without PKI, the worst thing that can happen compared to when FS-BGP is deployed, is that due to a VRS attack, at least one honest AS N_ℓ accepts at least one route $R = (N_{\ell-1}, \ldots, N_1)$ to some prefix P, such that for at least one honest intermediate AS N_i in R, subroute $(N_{i-1}, \ldots, N_1)$ is not N_i's the most preferred route to P, which would mean that the protocol does not converge due to a subroute consistency violation. Although requiring link-attestations diminishes the practical gains of having no PKI, having no PKI is still very practical and facilitates gradual, Internet-wide deployment of FS-BGP-PD as it relieves nodes of storing public keys of all other nodes and generating signatures for their every announcement. It also reduces communication overhead by getting rid of nodes' signatures in nodes' route announcements.

With respect to adversarial control of the flow of traffic on the Internet, Theorem 8.10 is a major milestone in understanding the security and efficiency tradeoffs that can be achieved in full versus no PKI deployment. Although with a VRS attack an adversary could cause an honest node to send traffic along an unintended route without that node's knowledge, the adversary could do the same without a VRS attack by simply diverting traffic to an unintended route of its choosing without that source's knowledge. The latter is an issue of data-plane accountability, and if the

Internet does not deploy a provably secure accountability protocol, e.g. [12, 34], then FS-BGP-PD with no PKI is just as good as with fully deployed PKI with respect to such an adversary. On the other hand, the only provably secure accountability protocols that are known to date require nodes to deploy a PKI or have shared keys, so having no PKI for FS-BGP-PD would yield no practical gains if the Internet does deploy a provably secure accountability protocol. Thus, in the beginning stages of partial deployment of secure path vector protocols, when there is no PKI, it may be more beneficial to deploy link certificates rather than have some nodes possess public keys but deploy no link certificates at all.

Currently, IETF [4] is considering standardizing a variant of S-BGP, called BGPSEC [42], that could work together with RPKI [5, 41]. RPKI consists of a hierarchy of authorities and AS's for certifying IP prefixes and AS numbers. Certificates for IP prefixes and AS numbers also contain certified public keys that are generated by the entities receiving the certificates. These keys would be used to run BGPSEC, and the results in this section apply to settings when either RPKI is partially deployed (i.e. not every AS gets a certificate for a prefix and a key) or RPKI is fully deployed but some AS's choose not to use their private keys to generate route attestations. Also, if the Internet were to be divided into some AS's that use S-BGP while the rest stick to BGP (partial deployment scenario considered in [31]), then our results with respect to S-BGP's and FS-BGP's security guarantees would apply only to each of the connected subgraphs of the Internet that choose to use S-BGP separately. To maintain overall Internet connectivity, AS's running S-BGP would have to use BGP when communicating with AS's that do not use S-BGP.

If origin authentication could be guaranteed with RPKI, then it is plausible that a similar, if not the same, hierarchy could be used to establish link certificates as is done in FS-BGP. We note, however, that if an adversary is allowed to corrupt various nodes in the RPKI and/or an analogous hierarchy for certifying communication links (i.e. entities that generate and/or certify keys, AS numbers, and communication links may be rogue), as we suggested in Section 5, to have well-defined, provable security guarantees in such scenarios, more sophisticated models and protocols would be needed to address rogue key and certificate attacks.

9 Conclusions and Future Work

We developed the framework for the provable-security treatment of path vector routing protocols. We defined an interdomain network, a path vector protocol and designed a formal security model for such protocols, which incorporates three general security requirements and is strong in terms of adversarial capabilities. Using our framework we analyzed security of the Secure BGP protocol. Assuming the underlying signature scheme is secure, we proved that S-BGP meets two out of the security definition's three requirements and showed how the protocol can be modified to meet all three security requirements at the same time. We also studied SoBGP and showed that it fails to meet two security goals. Finally, we studied security of partial PKI deployment when not all nodes have public keys. We investigated the possibilities of relaxing the PKI requirement while relying on non-traditional, non-cryptographic, physical security of interdomain networks such as the Internet, and we achieved possibly weaker, but still well-defined, notions of security. We also presented the necessary and sufficient conditions to achieve full security in the partial PKI deployment scenario.

As the Internet grows and evolves, so do its routing infrastructure and vulnerabilities. We believe our results fill the gap between the advances of modern cryptography and provable security methodology and practical networking protocols. Our framework should be useful for protocol

developers, standards bodies, and government agencies not only for verifying security guarantees of previous routing protocols, but also development and provable security analysis of future secure routing proposals.

One of the main criticisms of S-BGP is that it is very inefficient in terms of processing and communication overhead. There have been various proposals for more efficient ways route attestation mechanisms in S-BGP [36, 24, 21], and an important direction for future work would be to incorporate such proposals with our framework in order to design more practical and deployable secure routing protocols in a provably secure manner.

10 Acknowledgments

We thank Nick Feamster, Vytautas Valancius, Bogdan Warinschi and the anonymous reviewers for very useful comments. We also thank Mamta Upadhya for her participation in the early stages of the project.

References

[1] American Registry for Internet Numbers (ARIN). `https://www.arin.net/`.

[2] BGP Routing table analysis reports. `http://bgp.potaroo.net`.

[3] The Internet Assigned Numbers Authority (IANA). `http://www.iana.org/`.

[4] Internet Engineering Task Force (IETF) secure inter-domain routing group (SIDR). `http://datatracker.ietf.org/wg/sidr/charter/`.

[5] Resource Public Key Infrastructure (RPKI). `https://www.arin.net/resources/rpki.html`.

[6] C. Adams and S. Farrell. Internet X.509 Public Key Infrastructure: Certificate management protocols, 2004.

[7] B. Ager, N. Chatzis, A. Feldman, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a large European IXP. In *ACM SIGCOMM 2012*, Aug. 2012.

[8] W. Aiello, J. Ioannidis, and P. McDaniel. Origin authentication in interdomain routing. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 165–178, New York, NY, USA, 2003. ACM Press.

[9] G. Andersen, H. Balakrishnan, N. Feamster, T. Koponent, D. Moon, and S. Shenker. Accountable internet protocol (AIP). In *ACM SIGCOMM 2008*, Aug. 2008.

[10] M. Backes, I. Cervesato, A. D. Jaggard, A.Scedrov, and J.-K. Tsay. Cryptographically sound security proofs for basic and public-key Kerberos. In D. Gollmann, J. Meier, and A. Sabelfeld, editors, *ESORICS*, volume 4189 of *Lecture Notes in Computer Science*, pages 362–383. Springer, 2006.

[11] H. Ballani, P. Francis, and X. Zhang. A study of prefix hijacking and interception in the Internet. In *ACM SIGCOMM 2007*, Aug. 2007.

[12] B. Barak, S. Goldberg, and D. Xiao. Protocols and lower bounds for failure localization in the Internet. In *EUROCRYPT 2008*, Apr. 2008.

[13] A. Barbir, S. Murphy, and Y. Yang. Generic threats to routing protocols. *Network Working Group. IETF Request for Comments: 3962. Available at* `http://www.ietf.org/rfc/rfc4593.txt`, 2004.

[14] R. Barrett, S. V. Haar, and R. Whitestone. Routing snafu snips net service. Interactive Week, 1997. http://www.zdnet.com/zdnn/content/inwk/0413/inwk0032.html.

[15] M. Bellare, T. Kohno, and C. Namprempre. Authenticated encryption in SSH: provably fixing the SSH binary packet protocol. In *CCS '02*. ACM Press, 2002.

[16] S. M. Bellovin and E. R. Gansner. Using link cuts to attack internet routing. In *Tech. Rep., ATT Research, 2004, Work in Progress 2003 USENIX*, 2003.

[17] S. M. Bellovin, J. Ioannidis, and R. Bush. Position paper: Operational requirements for secured BGP. DHS Secure Routing Workshop, 2005.

[18] A. Boldyreva, M. Fischlin, A. Palacio, and B. Warinschi. A closer look at PKI: Security and efficiency. In T. Okamoto and X. Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 458–475. Springer, 2007.

[19] A. Boldyreva and V. Kumar. Extended abstract: Provable-security analysis of authenticated encryption in Kerberos. In *IEEE Symposium on Security and Privacy*, pages 92–100. IEEE Computer Society, 2007.

[20] A. Boldyreva and R. Lychev. Provable Security of (S-BGP) and other Path Vector Protocols: Model, Analysis, and Extenstions. In *ACM CCS 2012*.

[21] K. Brogle, S. Goldberg, and L. Reyzin. Sequential Aggregate Signatures with Lazy Verification from Trapdoor Permutations. In *ASIACRYPT 2012*.

[22] M. A. Brown. Renesys blog. Pakistan hijacks YouTube, 2008. http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml.

[23] K. Butler, T. Farley, P. McDaniel, and J. Rexford. A survey of BGP security issues and solutions. Technical Report TD-5UGJ33, AT&T Labs, 2004.

[24] K. Butler, P. McDaniel, and W. Aiello. Optimizing BGP security by exploiting path stability. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 298–310, New York, NY, USA, 2006. ACM Press.

[25] H. Chan, D. Dash, A. Perrig, and H. Zhang. Modeling adoptability of secure BGP protocols. In *SIGMETRICS '06/Performance '06: Proceedings of the joint international conference on Measurement and modeling of computer systems*, pages 389–390, New York, NY, USA, 2006. ACM Press.

[26] Y.-J. Chi, R. Oliveira, and L. Zhang. Cyclops: The internet as-level observatory. *ACM SIGCOMM Computer Communication Review*, 2008.

[27] J. Cowie. Renesys blog. China's 18-minute mystery, 2010. http://www.renesys.com/blog/2010/11/chinas-18-minute-mystery.shtml.

[28] J. P. Degarbiele and K. G. Paterson. On the (In)security of IPsec in MAC-then-encrypt configurations. In *ACM CCS 2010*.

[29] B. Dickson. Route Leaks – Requirements for Detection and Prevention thereof (v2). IETF Internet Draft, 2012. Available at http://tools.ietf.org/html/draft-dickson-sidr-route-leak-reqts-02.

[30] L. Gao and J. Rexford. Stable internet routing without global coordination. *SIGMETRICS Perform. Eval. Rev.*, 28:307–317, June 2000.

[31] P. Gill, M. Schapira, and S. Goldberg. Let the market drive deployment: A strategy for transitioning to BGP security. In *ACM SIGCOMM 2011*, Aug. 2011.

[32] S. Goldberg, S. Halevi, A. Jaggard, V. Ramachandran, and R. Wright. Rationality and traffic attraction: Incentives for honestly announcing paths in BGP. In *ACM SIGCOMM 2008*, Aug. 2008.

[33] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford. How secure are secure interdomain routing protocols? In *ACM SIGCOMM 2010*, Aug. 2010.

[34] S. Goldberg, D. Xiao, B. Barak, J. Rexford, and E. Tromer. Path-quality monitoring in the presence of adversaries. In *ACM SIGMETRICS 2008*, June 2008.

[35] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working around bgp: An incremental approach to improving security and accuracy in interdomain routing, 2003.

[36] Y. Hu, A. Perrig, and D. Johnson. Efficient security mechanisms for routing protocols, 2003.

[37] Y.-C. Hu, A. Perrig, and M. Sirbu. SPV: secure path vector routing for securing BGP. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 179–192, New York, NY, USA, 2004. ACM Press.

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

[39] S. Kent and K. Seo. Security architecture for the internet protocol. IETF RFC 4301, 2005. Available at `http://tools.ietf.org/html/rfc4301#page-4`.

[40] S. T. Kent, C. Lynn, J. Mikkelson, and K. Seo. Secure Border Gateway Protocol (S-BGP) - Real world performance and deployment issues. In *NDSS*. The Internet Society, 2000.

[41] M. Lepinski. An infrustructure to support secure internet routing.

[42] M. Lepinski. BGPSEC Protocol Specification (v4). IETF Internet Draft, 2012. Available at `http://tools.ietf.org/html/draft-ietf-sidr-bgpsec-protocol-04`.

[43] R. Mahajan, D. Wetherall, and T. Anderson. Understanding (BGP) misconfiguration. In *ACM SIGCOMM 2002*, Aug. 2002.

[44] A. Mityagin, S. Panjwani, and B. Raghavan. Analysis of the SPV secure routing protocol. Cryptology ePrint Archive, Report 2006/087, 2006. `http://eprint.iacr.org/`.

[45] S. Murphy. BGP security vulnerabilities analysis. *Network Working Group. IETF Request for Comments: 3962. Available at* `http://www.ietf.org/rfc/rfc4272.txt`, 2006.

[46] D. of Homeland Security. The national strategy to secure cyberspace, 2003. `http://www.whitehouse.gov/pcipb/`.

[47] U. of Oregon Route Views Project. `http://www.routeviews.org`.

[48] K. G. Paterson and G. J. Watson. Plaintext-dependent decryption: A formal security treatment of SSH-CTR. In H.Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 345–361. Springer, 2010.

[49] Y. Rikhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). *Network Working Group. IETF Request for Comments: 4271. Available at* `http://www.ietf.org/rfc/rfc4271.txt`, 2006.

[50] T. Ristenpart and S. Yilek. The power of proofs-of-possession: Securing multiparty signatures against rogue-key attacks. In M. Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 228–245. Springer, 2007.

[51] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz. Listen and whisper: Security mechanisms for BGP. In *Proc. First Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, Mar. 2004.

[52] S. Sundaresan, R. Lychev, and V. Valancius. Preventing attacks on BGP policies: One bit is enough. Technical Report GT-CS-11-07, Georgia Institute of Technology, 2011.

[53] The BGP TTL Security Hack. `http://tools.ietf.org/html/draft-gill-btsh-02`.

[54] T. Wan, E. Kranakis, and P. C. van Oorschot. Pretty secure BGP, psBGP. In *NDSS*. The Internet Society, 2005.

[55] R. White. Securing BGP through secure origin BGP. *The Internet Protocol Journal*, 6(3), Sept. 2003. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_6-3/ipj_6-3.pdf.

[56] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen. Scion: Scalability, control, and isolation on next-generation networks. *In Proceedings of IEEE Symposium on Security and Privacy (Oakland)*.

[57] M. Zhao, S. W. Smith, and D. M. Nicol. Aggregated path authentication for efficient BGP security. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 128–138, New York, NY, USA, 2005. ACM Press.

A SoBGP Definition and Security Analysis

In SoBGP [55], Origin Authorization Certificates are used to bind prefixes to certain AS's (just like address attestations in S-BGP) while AS Policy Certificates are used to allow nodes to learn of links and policies of remote nodes. Although similar to link attestations, AS Policy Certificates are not generated for communication links by a third trusted party; instead nodes (possibly corrupted) themselves disseminate their neighborship information. There is no equivalent of S-BGP Route Attestations in SoBGP, and this together with AS Plicy Certificates are the most essential differences between SoBGP and S-BGP. In this section we formally define SoBGP and show that, although it guarantees origin authentication, it does not guarantee route authentication and route validity.

Construction A.1. *Let* $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link})$, $\mathsf{Prefixes}$, $\mathsf{OrforPr}$, $\mathsf{relation}$, $\mathsf{preferto}$, $\mathsf{policy})$ *be an interdomain network, let* $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ *be a signature scheme with* $\mathsf{MsgSp} = \{0,1\}^*$*, and let* $\mathcal{CP}_s = (\mathsf{Kg_{CA}}, (\mathsf{CA}, \mathsf{U}), \mathsf{Vercert})$ *be the corresponding certification protocol as per Construction 2.1. In SoBGP* $= (\mathsf{Init}, \mathsf{An})$*, as part of* Init *the CA runs* $\mathsf{Kg_{CA}}(1^k)$ *to generate* $(pk_{\mathsf{CA}}, K_{\mathsf{CA}})$ *and each AS runs* $\mathsf{Kg}(1^k)$ *to generate* (pk, K)*.* An *is defined as follows.*

If node N_j*'s input* $\mathbf{P}_{N_j}$ *is nonempty (i.e.* $N_j \in \mathsf{Origins}$*), then for every prefix* $P \in \mathbf{P}_{N_j}$*,* N_j *does the following (note that this is just like in S-BGP as per Construction 4.1):*

- *CA and* N_j *interact according to* $(\mathsf{CA}, \mathsf{U})$*,* N_j *being* U*. The input to* U *is* $(pk_{\mathsf{CA}}, N_j, P)$*, the input to* CA *is* $(K_{\mathsf{CA}}, N_j, P)$ *and the outputs of both parties are* (N_j, P, cert)*. Origin Authorization* $OA^P_{N_j} \equiv \mathsf{cert}$ *is* N_j*'s certificate of ownership of* P*.*

- *Next, every* $N \in \mathsf{AS's}$ *runs* $\mathsf{Sign}(K_N(N, N', \mathsf{relation}(N, N')) = \sigma$*, for every* $N' \in \mathsf{Neighbors}(N)$*, to produce Policy Certificates* $PC_{NN'} \equiv (N, N', \mathsf{relation}(N, N'), \sigma)$ *that* N *makes publicly available to all other nodes in* $\mathsf{AS's}$*. Note that in our model of an interdomain network* policy *is publicly available, so AS Policy Certificates would need to be more sophisticated in scenarios where this is not true. In SoBGP, AS's may also be able to specify which other AS's their neighbors are allowed to export their routes, but we omit this detail as it is not essential to our security analysis of this protocol.*

For every new route announcement $(N_{j-1}, N_j, R = (N_{j-1}, \ldots, N_1), P, W, Aux = OA^P_{N_1})$ *that* N_j *receives from some neighbor* N_{j-1}*,* N_j *first performs origin authorization and policy certificate verification steps as follows.* N_j *runs* $\mathsf{Vercert}$ $(pk_{\mathsf{CA}}, N_1, P, OA^P_{N_1})$ *and outputs* $\perp$ *if the output of this computation is* 0*. Otherwise,* N_j *runs* $\mathsf{Ver}(pk_{N_i}, (N_i, N_{i+1},$
$\mathsf{relation}(N_i, N_{i+1}), PC_{N_iN_{i+1}})$*, for every* $1 \leq i \leq j-1$*, and outputs* $\perp$ *if at least one such computation*

outputs 0. *If none of the verification steps above results in* $\perp$, *then* N_j *performs the same operations as* N_j *would do in BGP upon receipt of* $(N_{j-1}, N_j, R, P, W, \varepsilon)$, *as per rules (1)-(3) specified in Section 4.1. Then, for every announcement* $(N_j, N_{j+1}, R', P, W', \varepsilon)$ *that* N_j *would send to* N_{j+1} *in BGP,* N_j *sends* $(N_j, N_{j+1}, R', P, W', Aux)$ *to* N_{j+1} *instead, where* $R' = (N_j, R)$.

If the underlying signature scheme $\mathcal{SS}$ is correct, the execution of SoBGP is the same as that of BGP in terms of how nodes update their routing tables and how they decide which routes to announce to their neighbors. Therefore, SoBGP is correct for the same classes of networks as BGP if the underlying signature scheme $\mathcal{SS}$ used to generate origin authorizations and policy certificates.

Theorem A.2. *SoBGP per Construction A.1 guarantees* origin authentication *for* $\mathcal{C}_0^{SoBGP}$ *if the underlying* $\mathcal{SS}$ *is uf-cma-secure.*

Proof. The proof follows from Theorem 2.2 and Lemma A.3 stated below. □

Lemma A.3. *Construction A.1 guarantees* origin authentication *for* $\mathcal{C}_0^{SoBGP}$ *if the underlying* $\mathcal{CP}$ *is uf-cda-secure.*

Proof. The proof is identical to that of Theorem 6.2 because the mechanism for achieving origin authentication in SoBGP with Origin Authorization Certificates is essentially identical to that in S-BGP with Address Attestations. □

Theorem A.4. *SoBGP per Construction A.1 does not guarantee* route authentication *for* $\mathcal{C}_0^{SoBGP}$.

Proof. The proof is essentially the same as that of Theorem 8.4, where the adversary causes subroute inconsistency with a VRS attack. This is because AS Policy Certificates in SoBGP (certifying physical communication links) do not guarantee route authentication by themselves for the same reason Link Attestations do not guarantee route authentication in S-BGP-PD by themselves when not every node has a public key. □

The following result points out the fact that link certification is not enough to guarantee route validity in general due to collusion when the certification is done by the nodes themselves.

Theorem A.5. *SoBGP as defined in Construction A.1 does not guarantee route validity for* $\mathcal{C}_0^{SoBGP}$.

Proof. (Sketch) The proof is very similar to that of 6.4. Here we present a specific example of an attack for the network in $\mathcal{C}_0^{\text{SoBGP}}$ depicted in Figure 2(a), where there is at least one valid route of length greater than three nodes. The adversary corrupts two non-neighboring nodes N_3 and N_8 that are on a valid route and creates a policy certificate for the fake link between them. There is nothing in SoBGP to prevent this from happening since both nodes are corrupted. The adversary then obtains the corresponding route announcement from the corrupted node closer to the origin N_3, and then propagates the corresponding route announcement on behalf of the other corrupted node N_8 to N_9. The route in the latter announcement is infeasible because the corrupted nodes are not actually neighbors, but there is no way to verify this fact by honest the honest node N_9 that is down-stream from the corrupted node farther away from the origin N_8. This is because the policy certificate of the fake link passes verification since it was created in a legitimate manner from the perspective of SoBGP. □

B Route Authenticity in the Restricted Version of S-BGP-PD

Construction B.1. *Let* $\mathcal{I} = (\mathbf{G} = (\mathsf{AS's}, \mathsf{link}), \mathsf{Prefixes}, \mathsf{OrforPr}, \mathsf{relation}, \mathsf{preferto}, \mathsf{policy})$ *be an interdomain network and* k *a security parameter. We define S-BGP-PD with an extra restriction (S-BGP-PDxR)* $= (\mathsf{Init}, \mathsf{An}')$ *as a path vector protocol identical to S-BGP-PD* $= (\mathsf{Init}, \mathsf{An})$ *but with the following restrictions in* An'*. When a node receives an announcement of a route from a neighbor, that node rejects that announcement if that route contains at least one node without a public key other than the neighbor sending the announcement. A node does not announce a route if it contains at least one node without a public key other than itself.*

Theorem B.2. *S-BGP-PDxR guarantees route authentication with* m*-PD for networks in* $\mathcal{C}_m^{S\text{-}BGP\text{-}PDxR}$*, for* $m \geq 1$*, if the underlying* $\mathcal{SS}$ *is uf-cma-secure and the Physical-Link-Security Relaxation holds (see Security Relaxation 1 in Section 8).*

Proof. This theorem trivially holds if $\mathcal{C}_m^{\text{S-BGP-PDxR}} = \emptyset$. Otherwise, let A be an efficient adversary attacking route authentication of S-BGP-PDxR for a network $\mathcal{I} \in \mathcal{C}_m^{\text{S-BGP-PDxR}}$ with $m \geq 1$ and $|\mathsf{AS's}| \geq 2$, whose description is polynomial in k. As a result of A's attack, $N_\ell \in \mathsf{Honest}$ accepts an announcement $(N_{\ell-1}, N_\ell, R = (N_{\ell-1}, \ldots, N_1), P, W, Aux)$, such that $\exists\ 1 \leq i \leq \ell - 1$ so that $N_i \in \mathsf{Honest}$ has never sent announcement $(N_i, N_{i+1}, R' = (N_i, N_{i-1}, \ldots, N_1), P,\ W', Aux')$, for some W', Aux', while Relaxation 1 holds.

Either N_i has a public key or it does not. If $N_i \in \mathsf{nopubk}$ and $i = \ell - 1$, then A must have either generated that announcement and sent it on behalf of N_i or intercepted and modified some other N_i's announcement. This cannot happen as it would violate the Physical-Link-Security Relaxation (see Security Relaxation 1 in Section 8), in which case A would not win. If $N_i \in \mathsf{nopubk}$ and $i < \ell - 1$, then R' must contain at least one node N_{i+1} between N_i and N_ℓ. However, since $N_\ell \in \mathsf{Honest}$, by construction of S-BGP-PDxR, it could not have accepted a route announcement with at least one node without a public key other than $N_{\ell-1}$, so we exclude this case from the proof. The only remaining option is that $N_i \notin \mathsf{nopubk}$. The rest of the proof is identical to that of condition (1) in Theorem 8.6. □

This theorem shows that route authentication can be guaranteed as long as only the last couple of nodes on routes do not have public keys. As was already pointed out in [31], these are the smaller networks that are likely to be at the edge of the Internet, i.e. stub networks (AS's that do not have any customers of their own such as small university and corporate networks). This result is still very significant since stub networks make up over 80% of the Internet [26].