Privacy Preserving IP Traceback

Le Su¹, Dinil Mon Divakaran², Vrizlynn Thing¹
¹ Fusionopolis Way, #21-01 Connexis (South Tower), Singapore 138632
{isu,vriz}@i2r.a-star.edu.sg
² Singtel, NUS-Singtel Cyber Security R&D Lab
I-Cube #03-47, 21 Heng Mui Keng Terrace, Singapore 119613
dinil.divakaran@singtel.com

Abstract

Tracing the source and path of traffic flows is an important problem that is useful in different network security and forensic solutions. Many solutions have been proposed for IP traceback in the past few decades, based on logging or marking, or a combination. Yet, there is no ubiquitously deployed traceback solution in the Internet. While scalability is the challenge facing logging-based approaches, marking-based approaches reveal sensitive information of ISP networks. In this work, we look into the problem of preserving the privacy of ISP networks in marking-based traceback solution. To this end, we propose the first privacy-preserving solution for IP traceback, that does not reveal the topological information of ISP networks, while still serves traceback queries. We present both numerical analysis and simulation-based studies, to evaluate the performance of our solution.

1. Introduction

Internet attacks have become a growing threat to the global Internet infrastructure, with recent DDoS attacks reaching staggering rate of 1 Tbps [1]. One basic step to prevent such DDoS attacks is to find the attack source and subsequently deploy preventive measure. However, there is a challenge: those source IP addresses are usually spoofed by the attackers in order to avoid successful identification.

IP traceback provides a tracing mechanism to reconstruct the traffic routing path, and possibly identify the attack origin. IP traceback is useful for attack deterrence, attack mitigation and forensic investigation: it also finds use in traffic path validation, bottleneck identification and fault diagnosis. Existing traceback solutions can, in general, be categorized into three types: marking-based, logging-based, and hybrid approaches. In marking-based solutions, routers embed traceback information in transiting traffic (flows or packets), consequently conveying the relevant information to end-hosts for path reconstruction. In logging-based solutions, relevant packet information are stored in either intermediate routers or at a designated storage service, for later inquiry and information retrieval in order to reconstruct the path. Hybrid solutions take advantages of both approaches, while reducing their negative effects.

While traceback solutions have matured over the years, there is no widely deployed solution in the Internet yet. Logging-based solutions are challenged by scalability: the storage capacity for logging traffic information needs to scale with network capacity. Marking-based solutions, on the other hand, leak sensitive private information of ISP (Internet Service Provider). Attackers and competitors can extract topology information of ISPs by sending traffic flows across different paths [3]. Information leak is of major concern to ISPs, as it can lead to attacks, loss of revenue (if competitors manipulate path selection of an ISP), etc.

In this context, a privacy-preserving traceback technique, which protects sensitive information, such as router identity and network topology, may well be a driving factor for ISPs to deploy such solutions in their networks. A trivial way to protect router identities during marking is to, instead embed router alias in packets, in a way that makes router identification difficult for others [8]. However, this is not a satisfactory solution as the alias is a rather static representation. By collecting sufficient marking information, an attacker could obtain partial and necessary information, such as the most traveled routers. This may potentially be dangerous: such routers once identified, could become primary targets.

In this work we introduce a privacy-preserving IP traceback mechanism: a marking-based solution which protects the identity of marking routers as well as the overall network
topology of participating ISPs. The main idea of our solution is to leverage on *keyed-hash message authentication code* (HMAC) approach, but using only a few bits from the output as router mark. While enabling efficient, privacy-preserving and accurate path reconstruction, our solution also gives the flexibility to an ISP, to decide on the number of marking bits independent of other ISPs. This gives ISPs control over the accuracy of router identification during the traceback query. Our contributions are: 1) to the best of our knowledge, this is the first work that addresses the open problem of preserving privacy in marking-based traceback; 2) we provide a secure and efficient solution to the privacy-preserving traceback problem, defining the marking as well as tracing algorithms; 3) we evaluate the proposed system through both numerical and simulation-based analysis to provide insights into the influence of system parameters.

We present the system architecture and the components of our privacy-preserving traceback solution in Section 4, and the algorithms for the marking and tracing processes in Section 5. Finally, we evaluate our solution in Section 6.

2. Related Works

One of the pioneering works in logging-based traceback is [12], wherein the authors proposed the hash-based IP traceback scheme that stores the digest of packets using Bloom filters. The disadvantage of logging-based solutions is the large storage overhead required at routers for packet logs. A hybrid traceback approach utilizes advantages from both marking and logging-based approaches, in order to reduce the number of packets required for path reconstruction, and the storage overheads at each intermediate router. We refer readers to [4] for a comprehensive review of both logging-based and hybrid traceback solutions. Below, we discuss marking-based traceback techniques.

Marking-based schemes could be categorized into two types: deterministic packet marking (DPM) and probabilistic packet marking (PPM). Usually for DPM, only the entry router will embed the router specific information into the packet, while for PPM, each router probabilistically selects the packet to be marked. During traceback, DPM could reconstruct the router path based on the information embedded in the packets, while for PPM it needs to collect sufficient amount of packets to complete the path reconstruction. Readers may refer to [2] for a representative work of DPM, [11] for a well-known work on PPM, and [5] for a more detailed summary of other works.

There are a few works that are partially related to privacy preservation in traceback, [6, 8] and [9] are some examples. The motivation of these works is to improve the efficiency of marking solution, by designing a more compact representation (using different kinds of Bloom Filters [10], Huffman coding [6], etc.) for embedding router identities in packet headers. Therefore, the solutions are not strictly privacy-preserving in nature; the representations can be seen as aliases. Aliases leak partial information about the network. For example, if the mapping between the router identification and IP address is static, by collecting sufficient marks, an attacker could easily deduce the path and even network topology. Changing the mapping between router identifications and IP addresses frequently, to avoid such attacks, may also be quite a hassle for both ISPs and routers.

Authors in [13] studied marking authentication problem utilizing the HMAC technique. But, instead of preserving the mark privacy, their solution focused on the mark integrity and authenticity. As the secret key is shared with the endhost, any user could reconstruct the path and the network topology by collecting sufficient number of marked packets.

3. Design Requirements

Identities of routers in an ISP that mark traffic should not be revealed to anyone other than ISP itself. Other entities, such as end-user or other ISPs, can exploit the information if the identities are exposed. Yet, the traceback solution should be able to identify the path taken by the traffic with high accuracy to assist forensic investigation, deploy dynamic mitigation solution, etc. Based on these, we put down the design requirements in a privacy-preserving traceback solution:

- **Anonymity:** with high probability, it is infeasible for an entity not in possession of secret information (e.g., secret key) to recover the identity of the marking router.
- **Indistinguishability:** without the secret key, it is infeasible for an entity to identity whether two or more marks are produced by the same router, or a particular mark was generated from which input.
- **Traceability:** a trusted entity in possession of the secret key can successfully identify the router that produces a particular mark with high probability.
- **Robustness:** it is difficult for an entity to produce a valid marking that is able to fool the tracing algorithm successfully, by collating the information gathered.
- **Scalability:** the solution should be reasonably scalable: it does not incur large storage or computational overhead to the entities involved—ISP, router, and user.
- **Independent execution:** it should not require interactive communication between router and ISP during marking and validation. This property ensures no additional communication cost due to the traceback solution.

4. Privacy-Preserving Traceback Solution

4.1. Architecture

For privacy-preserving traceback, we consider a typical two-layered architecture involving routers and their respective controlling ISPs. For each ISP, the routers under its
control could be categorized as edge routers and internal routers. For marking-based traceback solutions, it is impractical for each router along the routing path to participate in the marking process, as this will incur large overhead; neither is marking at all routers necessary. A more practical approach is to have only the edge routers mark the traffic of interest. Though marking can be packet-based or flow-based (where flow is set of temporally related packets due to the source and destination addresses, the ports and the protocol), we do not limit our design to any specific one. A single IPv4 packet header has 16 to 32 bits available for storing marks [7]. For a longer routing path, where the overall marking length exceeds the free space available in a single packet, the marking can be spread across multiple packets, utilizing piggyback rides (e.g., in [5]). IPv6 has larger free space, and thus it is easier to accommodate marks. In general, observe that, at flow-level, there is even more room available due to the multiple packets that compose a flow. However, in our work, we do not focus on mandating the marking length; instead, our analyses consider the impact of the number of bits in our design. Yet, for ease of exposition, from now on, we often refer to packets as the entities being marked. For a given packet, an edge router that marks the packet is either an ingress router or an egress router. As a packet transits an egress router, it hops from one ISP to another.

Our solution makes the following three assumptions: 1) all routers within an ISP are controlled by the same authority (the ISP). This authority trusts the routers its controls, the routers in the same ISP trust each other; 2) within an ISP, all marking routers are aware of each other’s identities; 3) the end-user, or destination, knows the last ISP in the path taken by a packet; normally it is the ISP the user subscribed to, for connection to the Internet.

We first provide a high level illustration for our solution. Fig. 1 illustrates the mark generation and aggregation process. Each router will generate its mark, append to the existing aggregated mark (from previous routers), and pass to the next one. When the packet traverses from one ISP to another, there will be a domain mark appended. In this example, router $R_3$ in ISP$_2$ generates the domain mark of ISP$_1$. Below, we describe the concept in detail.

4.2. Design concept

Designing a privacy-preserving IP traceback solution is challenging. Trivial approach of directly adapting existing cryptographic primitives are impractical, due to either large overhead or inability to meet certain requirements. In the following, we develop the important components and processes constituting our system traceback.

4.2.1 HMAC, a plausible candidate

Along with privacy protection, efficiency is also one of the key factors in designing the solution. HMAC becomes a reasonably acceptable candidate for three reasons: (i) it is extremely fast in computation; (ii) verification of output is easy and fast, if input is the same; and (iii) it is impractical (depending on the input size) to find or forge an input that results in the same output.

The first point allows practical deployment and acceptable low system delay when processing a packet. Second property supports identification of the marking router by an authority as long as the input information is correct. Last property prevents any unauthorized entity from exploiting the mark for possibly launching any attack. This property is further strengthened by the secret key used as the input for HMAC: any entity without the secret key will find it impractical to forge a mark that beats the validation process.

There exist such HMACs that are widely deployed, for example SHA-family based designs. However, the biggest challenge is that their output length is rather long (hundred of bits), making it unsuitable to be used as the exact packet mark. Our idea is to explore the possibility of extracting significantly smaller number of bits from the actual output, such that those extracted bits could still represent the marks of routers without resulting in severe conflicts.

4.2.2 Constructing the input

Another important aspect to consider is the input content for HMAC. We describe it below.

Router identifier: To uniquely identify a particular router, router-specific information must be included in the input, of which the router’s egress IP address (or in general, router identifier) is a perfect candidate.

Packet-specific information: It is insufficient to just include the router identifier as input, otherwise for different packets that transit through the same path, the resulting marks would be the same. This repeating mark will leak partial information of the topology: with sufficient marks gathered, through statistical analysis an attacker could identify the important edge routers. It is thus important to also include packet-specific information. In our solution, we extract part of the packet payload and use it in conjunction with the router identifier as input for HMAC.

Aggregated marks: As the marks are aggregated along the routing path, we further include the aggregated mark received by each router as part of the input for producing the next mark. This would create a chain effect: the validation of
a particular mark would be affected if any bit of the preceding aggregated mark has been altered. This will increase the difficulty for an attacker to successfully bypass the tracing process by manipulating the marking bits.

**Counter, or round index:** The mark generated by a router might conflict with marks generated by other routers (as we only extract a few bits from the HMAC output), thus a router might go for multiple rounds of the mark-generation process.

This conflict situation is explained in detail below. Therefore, we also add a counter, as part of the input.

### 4.2.3 Mark conflict management

For a specific packet, with the corresponding input constructed above, the HMAC produces an output. With extremely high probability, this output will differ from other outputs that are produced from different inputs. However as we only intend to use a few bits from the output as the actual mark, say the first $b$ bits, it is possible that the first $b$ bits of an output will conflict with the $b$-bit mark produced by another router from a different HMAC output. This will cause a problem for unique identification of the marking router during the traceback validation process. Thus in our design, the marking router only uses a mark that has no conflicts with others routers. If the marks result in a conflict, the marking router will recompute the output for another round by increasing the counter. This is repeated until the marking router produces a unique mark in comparison with others.

For practical reasons, we also limit the number of rounds. However, there is a pending issue—how can a router efficiently produce a mark that is guaranteed to be unique, without incurring communication cost with other routers?

**Secret key management:** One straightforward approach to avoid mark conflicts is to broadcast the particular packet information to other marking routers in the ISP. Each possible marking router could produce its own mark, share, and compare among routers; and subsequently recompute if necessary to avoid conflict. Obviously this is not a practical approach as it has large communication overhead, leading to noticeable system delay. Instead, we want the actual marking router to perform such a step and guarantee that the mark is unique through a non-interactive process. With such requirement, we let an ISP and all its routers share the same secret key: this is a reasonable assumption as routers within the same ISP should trust each other. This way, the actual marking router could easily compute all the marks that may be created by other marking routers (should the packet traverse through them), compare its mark with the marks of others, and recompute if there is a conflict. All these steps are done locally at the router where the mark is generated.

We emphasize that, the next marking router in the routing path of the packet will still perform the same set of tasks. Thus it will generate its own mark, but with the input now changed due to the new aggregated mark.

### 4.2.4 Marking of ISP domains

When a packet leaves one ISP to the next-hop ISP, it is important to add a domain mark, in addition to the router mark. This new mark, instead of embedding router information, will include the ISP identification, as part of the input. It is not difficult to understand the rationale behind such design: the router mark was produced by a secret key shared within each ISP. Only the ISP with the secret key could successfully validate the marking router. An ISP usually connects to multiple other service providers; therefore, whenever one ISP completes its own router validation, the domain mark can be used to identify which previous ISP is responsible for further tracing, and subsequently pass the remaining marks.

A domain mark for an ISP is computed by the ingress router of its next-hop ISP, as during the validation process, it is the next-hop ISP’s responsibility to identify the previous ISP, and pass the aggregated mark for further traceback.

### 4.3 Illustration

In this section, we illustrate the validation process, as well as false positive scenarios arising in our solution.

**Mark validation:** The mark validation process, or the traceback process, is similar to the mark generation, except that, it is the ISP that computes the possible marks generated by each router, to compare with the mark it has obtained. The last ISP in the path receives the entire aggregated mark from the end-host. Once a particular ISP has completed the router validation, it removes the marks produced by its routers from the aggregated mark it has received, and forwards the remaining to the ISP it has identified from the domain mark. In Fig. 2, the ISP has the current aggregated mark ending with 0111 (as indicated, in the center). It computes the four marks generated by $R_5 - R_8$, and compares the results against 0111. In this particular example there is a unique match (produced by $R_3$); therefore, the ISP concludes the marking router to be $R_3$. The ISP further peels off 0111 from the aggregated mark in order to identify the next marking router (under its control), or next ISP for validation. As the actual number of marking routers (in a path within an ISP network) is known to the ISP, this decision is deterministic.

**False positive:** During mark validation, false positive may be introduced, leading the ISP to attribute the mark to a wrong marking router. Fig. 3 illustrates such a situation. In the first round of mark computation, the marking router $R_1$ produces a mark which conflicts with $R_2$, while $R_3$ and $R_4$ produces their respective unique marks as $R_3 : 1000$ and $R_4 : 1011$. This conflict between $R_1$ and $R_2$ leads to the next round of computation at $R_1$. In round 2, $R_1$ produces a unique mark 1000, and appends it to the aggregated mark.

When an ISP initiates the validation process, it obtains the same result as $R_1$ has computed in round 1. However, $R_3$ has uniquely produced a mark 1000 in the first round, same as what $R_1$ has uniquely produced in round 2. As the ISP
has no information of how many rounds has been carried out during the marking process, this situation will falsely lead the ISP to conclude \( R_3 \) as the marking router. However, we stress that, a false positive at a particular hop along the path will not affect the validation of other hops, as the marks, once processed, are removed from the aggregation. Therefore, a mark that leads to false positive is also removed from aggregation, leaving the remaining unique mark intact.

There are two straightforward approaches to avoid false positives during validation. The first is to let the marking router generate the marks for all routers until all the outputs are unique, and then embeds the unique mark to the packet. Under this approach, during validation, a router only needs to repeat the process of generating unique marks. The disadvantage is that, it will without doubt incur additional rounds of computation compare to our proposed method. The second approach is to include the round index into the mark. However this not only increases the marking length, but also reveals the computed round to an observer. Therefore, we do not take these approaches; and instead develop our solution based on the mark conflict management approach.

### 4.4 Property analysis

Anonymity and indistinguishability are clearly met, as we employ HMAC and extract the mark from the output. With-
value as hash computation is rather fast, so that the error probability is negligible.

**Algorithm 1** generate_mark

**Input:** Router: \( R \), Packet: \( P \), Aggregation: \( M \), Round: \( \gamma \)

1: \( m \leftarrow \text{generate_mark}(R, P, M, \text{round}) \) \( \triangleright \) input message to \( H \)
2: \( m \leftarrow \text{extract}(out, b) \) \( \triangleright \) extract first \( b \) bits
3: \( m \leftarrow \text{extract}(out, b) \) \( \triangleright \) extract first \( b \) bits
4: \( \text{return} \ m \)

**Algorithm 2** router_marking

**Input:** Router: \( R \), Packet: \( P \), Aggregation mark: \( M \)

1: \( \gamma \leftarrow 1 \) \( \triangleright \) round \( \gamma \)
2: \( \text{while} \ \gamma \leq \text{LIMIT} \ \text{do} \)
3: \( m \leftarrow \text{generate_mark}(R, P, M, \text{round}) \)
4: \( \text{other_marks} \leftarrow \text{emptyList}() \)
5: \( \text{for} \ \text{each} \ R_j \in \text{self.routers}, \text{such that} \ R_j \neq R \)
6: \( m_j \leftarrow \text{generate_mark}(R_j, P, M, \text{round}) \)
7: \( \text{append}((\text{other_marks}, m_j)) \)
8: \( \text{end for} \)
9: \( \text{if} \ m \notin \text{other_marks} \) \( \triangleright \) mark is unique
10: \( \text{return} \ (M \parallel m) \) \( \triangleright \) aggregated mark
11: \( \text{end if} \)
12: \( \text{increment round} \)
13: \( \text{end while} \)
14: \( \text{Exit with error} \)

5.2. Validation algorithm

Validation, at the ISP level, does the reverse process of marking. The user or victim presents the packet and the aggregated mark \( M \) to the last hop ISP (its service provider), and the ISP will initiate the validation process.

Given a packet and an aggregated mark, the function \( \text{ISP validation} \) in Algorithm 3 is executed by an ISP, to check and find the routers (if any) that produced the mark. The variable \( \text{NumMarkRouters} \) denotes the number of marking routers in a path in an ISP. The algorithm then proceeds to identify the routers that had marked. The \( \text{while} \) loop in the algorithm executes three actions: (i) it extracts the last mark in the aggregation \( M \), and updates \( M \); (ii) it calls the function \( \text{identify router} \) to obtain the identifier of the router that generated the same mark, and (iii) the router identifier \( R \) is appended to the list of marked routers.

A router \( R \) that originally generated a mark for a given input during the forward marking process, will generate the same mark during validation, as the input remains the same. Therefore, if the function \( \text{identify router} \) can not identify a router that generated the mark (within a predefined round), then it means that the ISP did not generate this mark. In this case, the only possibility is that the ISP was falsely identified, either due to a false positive in identifying the domain mark or due to a forged mark used for validation.

The function \( \text{identify router} \) is given in Algorithm 4. If a router is not identified within \( \text{LIMIT} \) number of rounds (\( \text{while} \) loop), it is considered an error due to false identification of the ISP. The \( \text{for} \) loop finds the routers that generate given mark \( m \) in a round. If there is only such router (line 10), it is identified as the marking router.

**Algorithm 3** ISP validation

**Input:** Packet: \( P \), Aggregation mark: \( M \)

1: \( \text{NumMarkRouters} \leftarrow \text{number of marking routers in a path within the ISP} \)
2: \( \text{MarkedRouters} \leftarrow \text{emptyList}() \)
3: \( n \leftarrow 0 \)
4: \( \text{while} \ n < \text{NumMarkRouters} \) \( \text{do} \)
5: \( \text{mark} \leftarrow \text{extract}(M, b) \) \( \triangleright \) extract last \( b \) bits
6: \( M \leftarrow \text{stringRemove}(M, \text{mark}) \) \( \triangleright \) remove mark from aggregation \( M \)
7: \( R \leftarrow \text{identify router}(P, M, \text{mark}) \)
8: \( \text{append}((\text{MarkedRouters}, R)) \)
9: \( \text{increment} \ n \)
10: \( \text{end while} \)

**Algorithm 4** identify_router

**Input:** Packet: \( P \), Aggregation: \( M \), Mark: \( m \)

1: \( \gamma \leftarrow 1 \) \( \triangleright \) round \( \gamma \)
2: \( \text{while} \ \gamma \leq \text{LIMIT} \) \( \text{do} \)
3: \( \text{router list} \leftarrow \text{emptyList}() \) \( \triangleright \) initialize in each round
4: \( \text{for} \ \text{each} \ R \in \text{self.routers} \)
5: \( \hat{m} \leftarrow \text{generate_mark}(R, P, M, \gamma) \)
6: \( \text{if} \ \hat{m} = m \) \( \triangleright \) potential candidate
7: \( \text{append}((\text{router list}, R)) \)
8: \( \text{end if} \)
9: \( \text{end for} \)
10: \( \text{if} \ \text{length}((\text{router list}) \) \( = 1 \) \( \triangleright \) unique mark
11: \( \text{return} \ \text{router list}[1] \) \( \triangleright \) return the only element
12: \( \text{end if} \)
13: \( \text{increment} \ \gamma \) \( \triangleright \) else, compute another round
14: \( \text{end while} \)
15: \( \text{Exit with error} \)

6. Performance Analysis and Evaluation

6.1. Analysis of the marking process

Consider a packet \( P \) arriving at a router \( R_i \) of a specific ISP. The number of marking routers in different ISPs may well be different; but for our analysis here, we assume there are \( n \) marking routers in an ISP. To keep the architecture flexible, we assume that, at the time of arrival, a router is not aware which other routers would mark the packet. The router \( R_i \) generates a mark \( m_i \) for packet \( P \) using the secret key. \( R_i \) also generates marks for other routers using their identities (and the shared secret key), to check if there is any conflict between the mark \( m_i \) and any of the marks \( m_j, j \neq i \) of other routers. If so, router \( R_i \) ignores the current mark, and proceeds to generated a new set of marks in another round for all routers, until its mark does not conflict with others.
In a given round $\gamma$, the probability a mark generated by router $R$ conflicts with marks generated by other routers is:

$$P_{R,\gamma} = 1 - \left(1 - \frac{1}{2^b}\right)^{n-1} \quad (1)$$

The above probability is independent of router identity and the round in which the computation is performed. The probability of conflict at a router decreases exponentially in the number of marking bits. Fig. 4 plots Eq. (1) by varying the number of bits and the number of routers with marking roles. For an ISP network of moderate size around 10 marking routers, we observe that 10 marking bits are enough to keep the probability of conflict at a low value of 0.01.

### 6.2. Analysis of the validation process

In this section, we analyze the false positive probability of our system. Consider the scenario where an ISP receives the (possibly partial) aggregated mark, and proceeds to find which of its marking routers had marked the packet. We now describe the possibility of a false positive event.

When a router $R_i$ generates a unique mark $m_i^r = m$ for a packet $P$ at round $r$, it is guaranteed that the marks generated by other routers $R_j, j \neq i$ are different from $m$. However, it is possible that the mark $m$ was also generated by a router $R_j, j \neq i$ in one of the previous rounds $\gamma$, such that $\gamma < r$ (recall our false positive example in Section 4.3). Such a conflict that occurs across different rounds, results in incorrectly identifying a marking router in the validation process.

In this context, we define the false-positive probability $F$ as the probability that, for an arbitrary (input) packet $P$, any router $R_j, j \neq i$ generates the same mark $m$ in one of the rounds $\gamma, 1 \leq \gamma < r$, given that mark $m$ was uniquely generated by router $R_i$ in its $r^{th}$ round.

To derive the equation for $F$, let us consider each round preceding $r$. If $r = 1$, the false positive probability is zero because, by definition, $m$ is a unique mark in the round $r = 1$ (and there is no round preceding round 1). Next, consider the case where $m$ was uniquely produced in the second round ($r = 2$) by router $R_i$. To have a false positive, the mark $m$ should have been uniquely produced by a router $R_j, j \neq i$ in round $r = 1$. Therefore, the probability of producing a mark in round $r = 1$ that results in false positive, $P_f^1$, is:

$$P_f^1 = P_f = \frac{1}{2^b} \times \left(1 - \frac{1}{2^b}\right) \times \left(1 - \frac{1}{2^b}\right)^{n-2} \quad (2)$$

In Eq. (2), the first term is the probability of generating mark $m$ by one of the routers $R_j, j \neq i$. The second term is the probability that $R_i$ generates a mark $m$ different from $m$ in round $r = 1$, as $m$ generated by $R_j$ needs to be unique. For the rest of the routers, the marks produced should be such that none produces $m$, but at least one of them should produce a conflicting mark with $R_i$, i.e., $m_i$. The latter ensures that $R_i$ goes to next round due to conflict. This probability, for the remaining $n-2$ routers, is denoted by the third term.

We now generalize the probability of a round $\gamma, 1 \leq \gamma < r$, generating a mark that leads to a false positive event (when the original mark was produced in round $r$):

$$P_f^\gamma = \left\{ \begin{array}{ll}
\prod_{i=1}^{\gamma-1} (1 - P_f^i) & \text{if } \gamma = 1, \\
\prod_{i=1}^{\gamma-1} (1 - P_f^i) \times P_f^i & \text{if } \gamma > 1.
\end{array} \right\}$$

The equation for $P_f^\gamma$ has a recurrence form. Given that router $R_i$ was the actual router that generated the mark $m$ in round $r$, for a specific input packet, the probability of false positive,

$$F = \frac{1}{2^b} \left(1 - \frac{1}{2^b}\right)^{n-1} \times P_f^r \quad (3)$$

In Fig. 5, we plot the above equation for round $r = 3$. With increasing number of routers, the relative increase in false positive probability decreases. As observed, the false positive probability decreases exponentially in the number of bits. We also evaluated the false positive probability for increasing rounds and observed decrease in false positive probability, but the difference was negligible (and not plotted).

We developed a network simulator with the forward marking and reverse validation processes. All the important parameters, such as number of ISPs, number of routers in an ISP, number of marking routers in a path of an ISP, number of marking bits, etc. can be configured in this simulator. We use the simulator consisting of four ISPs to estimate the false positive rate. We fixed the number of routers in an ISP that eventually mark a packet to be two. The two routers were chosen randomly among the marking routers in an ISP, imitating random selection of a routing path. Note that, changing the number of marking routers in a path does not affect the results, as the conflict in each round and the false positive probability are affected by the number of marking routers in an ISP, which is greater than or equal to the actual number of routers that mark a packet along a path.

The number of possible marking routers in an ISP was set from the list $[5, 10, 15, 20]$; and in each such scenario, the number of marking bits was varied from 4 to 12 bits. For a given scenario and fixed number of marking bits, a total of 100,000 packets were simulated; and each setting was run 15 times, such that the 95% confidence interval is negligible (in comparison to the mean values plotted). The false positive rate is the fraction of packets for which there were false
positives. The results are plotted in Fig. 6. Each point on the figure is the mean false positive rate over 15 runs. We observe that the false positive rate decreases exponentially; it is negligible for an ISP with a marking length of 10 bits. With 8 bits marking length, the maximum false positive rate among all scenarios is extremely low at 0.55%. Note that Fig. 5 and Fig. 6 represent different set of experiments: the former reflects Eq. (3) with specific round $r$, while Fig. 6 is a full-fledged simulation of different scenarios with no limitation on the number of rounds.

Though the analyses above are on router mark conflicts and identification; the same analyses apply to ISP domain mark conflicts and ISP identification. However, because the number of ISPs an edge router peers with, is relatively much lower in comparison to the total number of peering ISPs, the conflicts and false positives are expected to be negligible.

### 7. Conclusion

In this work, we proposed a traceback solution that preserves the privacy of ISP networks. Our solution, based on HMAC, is fast and enjoys several nice properties essential for IP traceback. We developed the algorithms that form the marking and validation algorithms. Subsequently, we performed both numerical and simulation-based analyses to evaluate the performance of the proposed solutions. The results showed that the marks generated by the routers and ISPs can be validated during the traceback query with high accuracy. The accuracy is a function of the number of bits used for marking, and our design allows each ISP to choose the length of mark independently and different from the rest.

### Acknowledgment

This work is supported by the Singapore National Research Foundation NCR Award No. NRF2014NCR-NCR001-034.

### References


