Peer-to-Peer Anonymous Networking Using Coding

Christopher S. Chang[1], Tracey Ho[2], and Michelle Effros[2]

***Abstract*— In this paper, we consider design and analysis of coding-based anonymous routing systems in peer-to-peer (P2P) overlay networks. An unknown subset of participating nodes is adversarial, and can collude to try to identify the communicating nodes through passive observations. The first part of this work considers subgraph setup in the absence of a reliable public key infrastructure (PKI). As in the "slicing the onion" scheme proposed by Katti et al., a sender constructs a layered subgraph over which coding is performed, but we propose a new coding scheme with a formal information theoretic security characterization. We consider optimization of protocol parameters to maximize the adversary's uncertainty, as measured by the entropy of the source and sink identities, and show that a randomized strategy can improve anonymity and resource usage efficiency. The second part of this work focuses on the data transmission phase, assuming availability of a subgraph setup scheme (either PKI-based or coding-based) and end-to-end encryption. We use network coding at intermediate nodes to improve networking performance and reduce complexity by replacing expensive cryptographic operations at each hop with simpler linear algebra operations.**

I. Introduction

The goal of anonymous networking is to hide the identities of communicating nodes. Applications of anonymous networking include electronic voting, military communications, and communications of a sensitive commercial or political nature.

Many anonymous networking systems that rely on a public key infrastructure (PKI) have been proposed, starting from the seminal work of Chaum [1] on mix networks, to the "onion routing" approach of Reed et al. [2] and the Tor protocol [3] which is the most widely used anonymous networking system currently. The public keys of intermediate relay nodes are used to recursively encrypt information at the source, and each relay node decrypts a layer of information using its private key. A number of anonymous networking proposals have also focused on P2P overlay networks, as the decentralized nature of P2P systems and their potential to scale to a large number of participating nodes are attractive for various scenarios. Such proposed schemes include Tarzan [4] (a P2P networking system based on onion routing), MorphMix [5] (similar to Tarzan but where the routes are determined by intermediaries), Salsa [6] and Torsk [7] (structured approaches to build scalable P2P anonymous networks), which require a reliable public key infrastructure.

Other P2P anonymous networking schemes such as Crowds [8], AP3 [9], and the "slicing the onion" scheme of Katti *et al*. [10] have less reliance on a PKI, and are useful in situations where a PKI is not available/reliable or may potentially be compromised. Crowds and AP3 use randomized forwarding, and protect the sender but not the receiver identity. The "slicing the onion" scheme considers both sender and receiver anonymity. It splits routing information across multiple relay nodes which are arranged in a rectangular subgraph consisting of l layers of d nodes each, as illustrated in Fig. 1. Information intended for each node (which includes information about its next hop nodes) is split into d slices and multiplied with an invertible $d \times d$ matrix. Thus, each node is able to decode its intended information using the packets received from all nodes in the previous layer, while being unable to decode information intended for other nodes. The type of security provided is information theoretic in nature, relying on path diversity and not on computational assumptions. However, the scheme is not strongly secure in the information theoretic sense (as measured by mutual information) and no formal analytical characterization of the security of the scheme is given, e.g., it is not clear how much information is leaked to an adversary who controls a subset of relay nodes. In Section III we propose a different coding scheme over the subgraph with a formal information theoretic security characterization, and further consider optimization and randomization of the subgraph parameters.

Several recent works [11]–[13] have investigated the use of network coding in anonymous networking assuming availability of a separate scheme such as those discussed above, for setting up a subgraph anonymously. These works propose modifications to conventional network coding to protect the coded packets against traffic content correlation. In practical network coding, the source information is divided into multiple generations of packets. Network nodes carry out random linear coding among packets of each generation, and the coding operations are captured by global encoding vectors (GEVs) that undergo the same linear coding operations as the data. Such coding is not compatible with the layered encryption schemes employed in non-network coded anonymity schemes to cryptographically transform

This material is based upon work supported by the Defense Advanced Research Project Agency (DARPA) and Space and Naval Warfare Systems Center Pacific under Contract No. N66001-11-C-4003. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Project Agency and Space and Naval Warfare Systems Center Pacific.

[1] C. Chang was with the Department of Electrical Engineering, California Institute of Technology, Pasadena CA 91125 USA. He is now with Samsung Electronics Co., Ltd., Seoul, South Korea (e-mail: csw.chang@samsung.com)

[2] T. Ho and M. Effros are with the Department of Electrical Engineering, California Institute of Technology, Pasadena CA 91125 USA (e-mail: {tho, effros}@caltech.edu)

packet contents at each hop. To address this issue, Fan *et al*. [11] proposed a scheme in which GEVs are encrypted using homomorphic encryption so that only the sink node with the appropriate decryption key can decode the GEVs and hence the message. While traffic content correlation is made more difficult for the adversary, an adversary who controls multiple participating nodes can still check if a packet is in the span of another set of h packets with $O(h^3 + hn)$ complexity, where n is the length of each packet. In Section IV we propose an alternative approach using algebraic coding over layered subgraphs, where the complexity of such content correlation attacks is substantially higher. Wang *et al*. [13] proposed a lower overhead network coding scheme where only routing information, flow and generation numbers are encrypted while GEVs and message contents are not encrypted. The scheme hides the correlation of upstream and downstream GEVs of flows by designing the GEVs to be linearly dependent with those from other flows, but is only secure against external observers and not internal participating nodes. Gasti *et al*. [12] considered the problem of checking data integrity in anonymous network coded peer-to-peer file sharing networks in the presence of active adversaries that may corrupt coded packets. Unlike PKI-based integrity checking schemes used in the non-anonymous case, the authors proposed a hash-based approach for integrity checking of packets.

II. Model

The P2P overlay network consists of N participating nodes, each of which is adversarial independently with probability p. There are multiple concurrent unicast sessions, each with one source and one sink. Each source chooses a random subset of nodes from the network to construct an overlay subgraph, which it uses to communicate anonymously with an intended sink node. By choosing the nodes randomly, we avoid potential attacks where the adversary can try to bias the choice towards adversarial nodes by advertising favorable characteristics.[1] A relaying node can serve in multiple communication sessions (i.e., different subgraphs) simultaneously. We assume that the underlying physical network is generally well-connected so that there is path diversity between source and sink nodes.[2] As in [9], techniques from structured P2P overlay networks, e.g., [14], can be used to provide an efficient means of choosing a random subset of nodes from a large network, in conjunction with techniques for defending against Sybil attacks either with or without a PKI, e.g., [15].

We consider passive attacks from adversarial participating nodes, who collude to try to determine the source and sink identities from their observed transmissions and connectivity information. Each adversarial node is assumed to follow the protocol; we do not consider active attacks such as corruption or dropping of packets by relay nodes.

We use conditional entropy to measure anonymity, as in [16], [17]. Specifically, let $\mathcal{A}$ be the adversary's observations (i.e. observed messages and local connectivity) corresponding to a realization of adversarial node locations in the subgraph. We consider the conditional entropy $(S, T|\mathcal{A})$ of the source layer nodes S and the sink T given $\mathcal{A}$:

$$H(S,T|\mathcal{A}) = \sum_a \mathcal{P}(a) H(S,T|\mathcal{A}=a) \tag{1}$$

where $\mathcal{P}(a)$ denotes the probability of a particular adversarial realization $\mathcal{A} = a$.

III. Subgraph Construction Phase

In this section we focus on the subgraph construction phase. We propose a scheme that uses coding, rather than a PKI, to enable a source node to anonymously set up a subgraph and send a small secret message (e.g., a cryptographic key) to the sink.

As in the "slicing the onion" scheme [10] described in Section I, we consider a rectangular layered subgraph consisting of l layers of d nodes each.[3] The overlay links between nodes in two consecutive layers form a complete bipartite graph, and the coding scheme allows each node to decode its next hop routing information only if it receives messages from all of its neighbors in the previous layer. There are no overlay links between nodes that are not in successive layers. To prevent nodes from deducing information about their position in the subgraph from their in-degree or out-degree, the source sends from d distinct IP addresses.

While the subgraph structure is similar, our coding scheme differs from that of [10], and we provide a formal information theoretic security characterization. In particular, we show that as long as the adversary does not control a complete cut between the source and sink, the adversary gains no information about the connections in the subgraph other than the one-hop connectivity information provided to each node to specify its operation. Additionally, we consider optimization of the subgraph parameters and show that randomization of these parameters can improve the anonymity of the system and the resource utilization at the same time. As in Tor, rendezvous points for hidden services can also be set up by using our scheme to set up a subgraph from the source to one or more rendezvous nodes. The reverse path setup details can be found in [18, Chapter 3].

A. Coding Scheme

Let the layers of the subgraph be indexed in increasing topological order starting from the source layer. Consider nodes $\{u_1, \ldots, u_d\}$, $\{v_1, \ldots, v_d\}$, and $\{w_1, \ldots, w_d\}$ in three successive layers $k-1$, k, and $k+1$, respectively. A node

[1] In cases where information such as the geographic location of nodes provides an indicator of their probability of being adversarial, such information can be taken into account in the choice of relay nodes and subgraph design. This is a topic of future work.

[2] As will be evident from our study below, if all overlay paths between a source-sink pair pass through a very small number of physical nodes, then the anonymity of the coding-based schemes will be reduced.

[3] The results of this subsection generalize straightforwardly to any (not necessarily rectangular) layered subgraph where links between nodes in two consecutive layers form a bipartite graph, though, as noted above, the uniformity of node in-degrees and out-degrees is useful to prevent nodes from deducing information about their position in the subgraph from their in-degree or out-degree.

v_j $(j = 1, \ldots, d)$ has upstream neighbor nodes u_i $(i = 1, \ldots, d)$ and downstream neighbor nodes w_i $(i = 1, \ldots, d)$. Let the packet going from node x to y be represented by a vector g_x^y of symbols from a finite field $\mathbb{F}_q$.

The message intended for x consists of, in order, a last-hop flag ψ_x, a sink-flag ϕ_x, a secret θ_x for x (e.g., cryptographic key) if it is a sink, and packets to be forwarded further. The last-hop flag indicates whether the node is located at the end of the subgraph. The sink-flag indicates whether a node is a sink and has a secret θ_x intended to it; if it is not not a sink, θ_x consists of random symbols and contains no valid information. To simplify notation, let h_x denote the private information for node x, that is, (last-hop flag, sink-flag, secret) (i.e., $h_x \triangleq (\psi_x, \phi_x, \theta_x)$). The protocol ensures that each node v_j can decode its message $(h_{v_j}, g_{v_j}^{w_1}, \ldots, g_{v_j}^{w_d})$ by summing together the contents of all its received packets $(g_{u_1}^{v_j}, \ldots, g_{u_d}^{v_j})$.

The packet contents are defined recursively as follows. If k is the last layer ($k = l$), then $\psi_j = 1$ and v_j does not have any outgoing packets. Therefore, the message for v_j consists only of the private information for v_j. The contents of the packets transmitted from layer $k-1$ to k are defined as

$$\begin{aligned} g_{u_i}^{v_j} &= \left[v_j,\ K_{u_i}^{v_j}\right],\ i = 1, \ldots, d-1 \\ g_{u_d}^{v_j} &= \left[v_j,\ h_{v_j} - \textstyle\sum_{i=1}^{d-1} K_{u_i}^{v_j}\right] \end{aligned}$$

where each $K_{u_i}^{v_j}$ is an independent and uniformly distributed random vector of symbols from $\mathbb{F}_q$ of length equal to the message h_{v_j}. If k is not the last layer ($k \neq l$), $\psi_j = 0$ and we define the packet contents recursively based on the previous layer:

$$\begin{aligned} g_{u_i}^{v_j} &= \left[v_j,\ K_{u_i}^{v_j}\right],\ i = 1, \ldots, d-1 \\ g_{u_d}^{v_j} &= \left[v_j,\ (h_{v_j}, g_{v_j}^{w_1}, \ldots, g_{v_j}^{w_d}) - \textstyle\sum_{i=1}^{d-1} K_{u_i}^{v_j}\right] \end{aligned}$$

where each $K_{u_i}^{v_j}$ is an independent and uniformly distributed random vector of symbols from $\mathbb{F}_q$ of length equal to the message $(h_{v_j}, g_{v_j}^{w_1}, \ldots, g_{v_j}^{w_d})$. Fig. 1 is an example subgraph illustrating this construction.

Each node v_j in the network strips off its ID from each received packet $g_{u_i}^{v_j}$ and sums over the packets' contents to decode its message: $\sum_{i=1}^{d-1} K_{u_i}^{v_j} + (h_{v_j}, g_{v_j}^{w_1}, \ldots, g_{v_j}^{w_d}) - \sum_{i=1}^{d-1} K_{u_i}^{v_j} = (h_{v_j}, g_{v_j}^{w_1}, \ldots, g_{v_j}^{w_d})$.

Note that the size of the packet contents decreases with distance from the source. To prevent adversaries from deducing their location within the subgraph based on packet size, we maintain a constant packet size by padding with random symbols. The details of the padding algorithm are omitted due to limited space. It can be found in [18, Chapter 3].

Now, we characterize the information theoretic security properties of the signaling scheme against adversarial overlay nodes and overlay links, i.e. paths between overlay nodes that contain an adversarial physical node. We show that if there is a non-adversarial path (or equivalently, no adversarial cut) between the source and the last layer, the signaling is information theoretically secure in that colluding adversarial nodes obtain no information about the subgraph other than their own local connectivity.

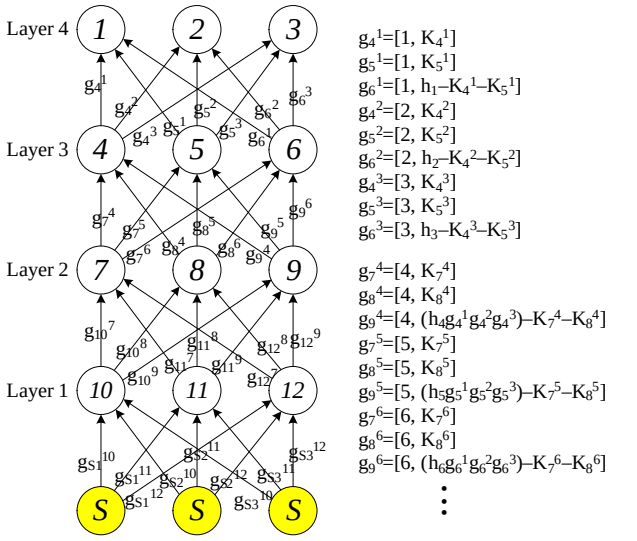

Fig. 1. Example of a rectangular subgraph with length of 4 and width of 3. g_i^j represents the packet from node i to j. K_i^j denotes an independent random vector of symbols from $\mathbb{F}$. The message consists of (sink-flag ϕ_j, last-hop flag ψ_j, the secret θ_j, and the packets to be forwarded), where $h_j = (\phi_j, \psi_j, \theta_j)$. Note that $\psi_i = 1$ if $i = 1, 2, 3$ and $\psi_i = 0$ otherwise.

Lemma 1. *For a uniformly distributed random vector $X \in \mathbb{F}_q^l$ independent from (Y, Z) where Y is an arbitrary vector from $\mathbb{F}_q^l$ and Z is a random vector, $X+Y$ is also a uniformly distributed random vector independent from (Y, Z).*

Proof. For any $a, b \in \mathbb{F}_q^l$ and $c \in \mathbb{F}_q^{l'}$,

$$\begin{aligned} & Pr\left[X + Y = b,\ Y = a,\ Z = c\right] \\ &= Pr\left[X = b - a,\ Y = a,\ Z = c\right] \\ &= Pr\left[Y = a,\ Z = c\right] Pr\left[X = b - a\right] \quad \text{since } X \perp (Y, Z) \\ &= Pr\left[Y = a,\ Z = c\right] \frac{1}{q^l} \quad \text{since } X \sim \text{Uniform Distr.} \end{aligned}$$

Now, we will show that $X + Y$ is an uniform random vector from $\mathbb{F}_q^l$.

$$\begin{aligned} & Pr\left[X + Y = b\right] \\ &= \sum_{i \in \mathbb{F}_q} Pr\left[Y = i\right] Pr\left[X = b - i\right] \\ &= \sum_{i \in \mathbb{F}_q} Pr\left[Y = i\right] \frac{1}{q^l} \\ &= \frac{1}{q^l} \end{aligned}$$

Therefore, we have $Pr\left[X + Y = b, Y = a, Z = c\right] = Pr\left[X + Y = b\right] Pr\left[Y = a,\ Z = c\right]$. □

Theorem 1. *As long as the adversary does not control a complete cut between the source and the last layer, the combined set of packets observed by the adversarial nodes does not reveal any information about the subgraph beyond the information intended for each of the nodes, i.e. one-hop connectivity and private information.*

Proof. The packet g_x^y from x to y contains the next-hop ID y and a payload represented by a vector f_x^y (i.e., $g_x^y = [y, f_x^y]$). The next-hop ID in the packet g reveals only local (one-hop) connectivity that is directly connected to the node. Therefore, we focus on the payload f in this proof. Let the set of payloads of outgoing and incoming packets of a node v be denoted by O_v and I_v, respectively. Note that I_v consists of independent uniform random variables K_w^v and a linear combination of the random variables and the message for v. The message for v is in turn composed of h_v and O_v with next-hop ID's of forwarding packets, where $h_v = (\phi_v, \psi_v, \theta_v)$ is the local information for v. Therefore, I_v does not reveal any further information about connectivity beyond one-hop from the adversaries than O_v (i.e., $\mathbf{I}(\text{connectivity}; I_v) = \mathbf{I}(\text{connectivity}; O_v)$).

We assume that adversaries (both nodes and links) do not form an edge-cut. Suppose that there is a trusted node v receiving and decoding message f_v. We will show $H(f_v|M) = H(f_v)$, where M is a set of messages collected from adversarial nodes and adversarial links.

Let S_k denote the set of adversarial nodes in layer k. Let s_k denote the number of symbols in $O_{v_i^k}$ where v_i^k is the i^{th} node in layer k (s_k are constant for all nodes in a layer). The worst case in which no edge-cut exists is that there exists a single path from the source to the last layer composed of non-adversarial nodes and edges, while all other nodes and edges are adversarial.

First, we consider the case where in each layer k, nodes $S_k = \{v_1^k, \ldots, v_{d-1}^k\}$ are adversarial. In this case, nodes $\{v_d^k : k \in [1,\ l]\}$ are not adversarial and form a trusted path between the source and the sink. It follows from the definition of the protocol that the vector $(O_{v_i^k} : v_i^k \in S_k, i \in [1,\ d], k \in [1,\ l])$ is independently and uniformly distributed over $\mathbb{F}_q^{\sum_{k=1}^{l} s_k |S_k|}$ and independent from f_v (i.e., all the adversarial messages are uniform i.i.d.).

Now, we consider the general case where the adversarial set S_k can include the node v_d^k. We will show that substituting v_d^k for some node v_a^k ($a < d$) in S_k does not change the distribution of $(O_{v_i^m} : v_i^m \in S_m, i \in [1,\ d], m \in [1,\ l])$ and its independence with f_v. The outgoing messages of a node v_d^k is $O_{v_d^k} = \{f_{v_j^{k+1}} - \sum_{i=1}^{d-1} K_{v_j^k}^{v_i^{k+1}} : j \in [1,\ d]\}$. We simplify the notation by dropping the layer indexes as $f_j - \sum_{i=1}^{d-1} K_i^j$ for $j \in [1,\ d]$. Applying Lemma 1 with $X = (\sum_{i=1}^{d-1} K_i^1, \ldots, \sum_{i=1}^{d-1} K_i^d)$, $Y = (f_1, \ldots, f_d)$, and $Z = (O_{v_i^m} : v_i^m \in S_m, i \in [1,\ d], m \in [1,\ l], (i,m) \neq (a,k))$, we have that $(f_1 - \sum_{i=1}^{d-1} K_i^1, \ldots, f_d - \sum_{i=1}^{d-1} K_i^d)$ is a uniform random vector, independent from $(f_1, \ldots, f_d)$ and $(O_{v_i^m} : v_i^m \in S_m, i \in [1,\ d], m \in [1,\ l], (i,m) \neq (a,k))$. We can proceed recursively in a similar manner to replace $v_d^{k'}$ for some node $v_{a'}^{k'}$ ($a' < d$) in $S_{k'}$ for other layers k'. □

B. Calculation of Conditional Entropy

In the previous section, we showed that if the adversary does not control a complete cut between the source and the last layer, the adversaries' observed packets do not reveal any information about the subgraph beyond their local connectivity. We assume a sufficiently large number of communication sessions (by sending cover traffic if necessary) so that adversaries that are not directly connected to each other do not know if they are in the same subgraph. This means that only connected adversaries can effectively collude to identify a source-sink pair. Therefore, in our calculation of the entropy metric (1) we only consider information revealed in the case of adversarial cuts, and deductions based on the adversary's local connectivity when there is no adversarial cut.

Our calculation considers three cases corresponding to different configurations of adversarial nodes. The calculation detail is omitted due to limited space. It can be found in [18, Chapter 3]. (a) Adversaries contain a vertex-cut of the network, possibly connected to an adversarial path upstream of the cut. In this case, they can identify all downstream nodes from the vertex-cut, and know the identities of nodes connected to the adversarial path. If there are multiple adversarial vertex-cuts in the network, we focus on the one closest to the source node, since it can identify more nodes including all the other vertex-cuts. (b) If adversaries control a path rather than a vertex-cut, they still obtain information about the identities of nodes in the subgraph to which they are connected; these nodes have more likelihood of being the sink compared to other nodes that are not known to be in the subgraph. (c) Lastly, adversaries can control multiple disconnected components (vertex-cuts and/or paths). If some paths or other vertex-cuts exist in the downstream network of a vertex-cut, they can be ignored in the calculation since they are identified by the vertex-cut closest to the source layer. We take into account a path only if it locates in the upstream network of a vertex-cut if any exists. For each, we calculate $H(S,T|\mathcal{A})$ and compare to pick the isolated adversaries set providing the smallest $H(S,T|\mathcal{A})$.

C. Experimental Results

We use the above analysis to calculate how anonymity varies with network and subgraph parameters. In Fig. 2, we plot conditional entropies normalized by the maximum possible uncertainty of source-sink pair, that is $\log_2[N(N-1)]$. For probability $p > 0.01$, the best subgraph shapes is $(l,d) = (8,3)$. In our simulations we observe that the optimal shape for a given subgraph size is generally invariant with adversarial probability p and network size N.

Note that small l and large d makes it more likely that the adversarial path length is relatively large compared to the subgraph length, reducing the uncertainty of the source layer. On the other hand, small d and large l makes it more likely to have an adversarial vertex cut, reducing the uncertainty of the sink. We observe from simulations that for the subgraph length of interest ($l \leq 20$), optimal width is either 3 or 4 in most cases, regardless of N and p.

We find that a useful heuristic is as follows. For a rectangle-shape subgraph, the optimal parameters (length and width) can be calculated approximately as the values for which the probability of an adversarial vertex cut intersects with the probability of an adversarial path of length at least

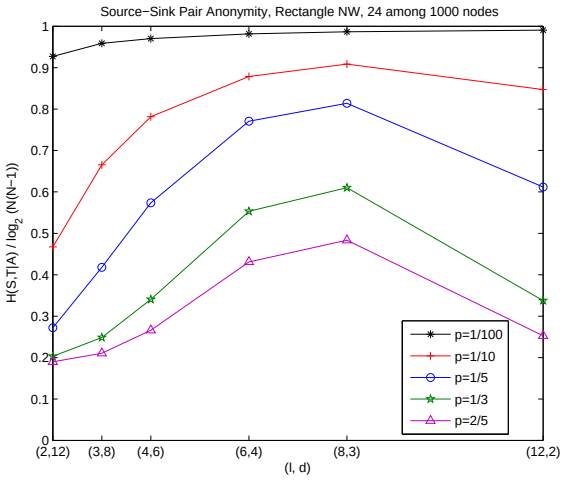

Fig. 2. Source-sink pair entropy conditioned on adversaries realization. Sink is randomly located within the subgraph. The subgraph contains 24 nodes randomly selected out of 1000 nodes. Plot of $\frac{H(S,T|\mathcal{A})}{\log_2[N(N-1)]}$ for different (l,d) and p.

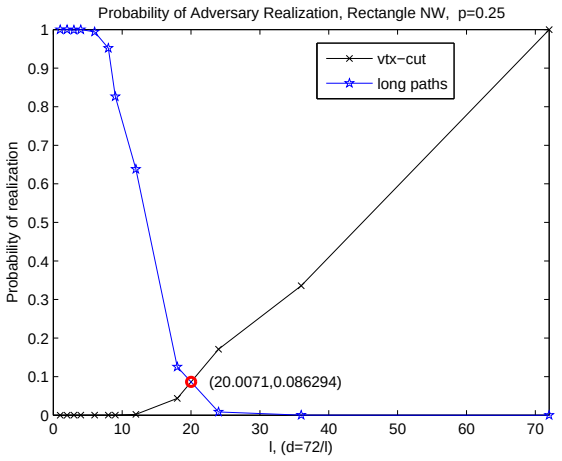

Fig. 3. Optimal subgraph parameters search for given subgraph size $l \times d = 72$ and compromising probability $p = 0.25$. The probability of vertex-cut and the probability of long-path are close near $l = 20$. The heuristic predicts that the optimal length is close to the intersection (either 18 or 24). The actual l_{opt} is 24.

half of the subgraph length, a condition which we refer to as "long-path". Thus, we choose the length to be one of the two divisors of the subgraph size closest to the intersection. The probability of an adversarial vertex cut is non-decreasing function of subgraph lengths whereas the probability of an adversarial long path is non-increasing function of subgraph lengths. In this context, vertex-cut can contain a connected path as before and long-path can contain a vertex-cut. An illustration is given in Fig. 3.

Simulations with different adversarial probabilities, network sizes, and subgraph sizes show that the heuristic correctly predicts the optimal subgraph shape in most cases. There exist some exceptional cases, though in these cases, the difference between entropies from the optimal subgraph shape and the entropies from the subgraph chosen by heuristic is very small. In most cases, the percentage difference is less than 0.5%.

D. Randomized Subgraphs

In this section, we propose a randomized strategy where the constructed subgraph has length drawn randomly from a suitable distribution. Note that randomizing width is not helpful since all nodes know the width from in-degree and out-degree of flows. We analyze the entropy metric in this case, and show by simulations that randomized length simultaneously provides better anonymity (higher entropies) as well as more efficient resource usage (shorter expected length of subgraph and smaller expected subgraph size) as shown in Fig 4. For the deterministic case where length l is fixed, we obtain $H(S,T|L=l,\mathcal{A})$ using the calculation in Section III-B. For the randomized case, we calculate $H(S,T|\mathcal{A})$ as the sum of three terms which we analyze separately:

$$H(S,T|\mathcal{A}) = H(L|\mathcal{A}) + H(S,T|L,\mathcal{A}) - H(L|S,T,\mathcal{A})$$

Similarly to the entropy calculation for the deterministic case, this calculation considers various cases for the relative positions of the sink and adversarial nodes. The details can be found in [18, Chapter 3].

We evaluate the performance of the randomized scheme by simulation experiments, in which the total number of nodes in the network N, adversarial probability p, and subgraph length set $\mathcal{L}$ are given, while we vary the distribution of lengths $P(l),\ l \in \mathcal{L}$ and the subgraph width d. In each simulation, $N = 10000$, p is fixed among $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, and d is fixed among $\{2,3,4,5\}$. The source first chooses a subgraph length from $\mathcal{L} = \{5,6,7,8,9,10\}$ with respect to some given probability distribution $P(l)$, and then constructs a subgraph. In Fig. 4, we plot normalized entropies conditioned on adversaries realization. Each rectangle represents a group of a same adversarial probability p, within which subgraph widths vary—one of $\{2,3,4,5\}$. The red star and black circle markers represent deterministic cases with fixed subgraph length $l = 10$ and $l = 9$, respectively. The blue square marker represents a random case with probability distribution $P(l) = \{\frac{1}{32}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}\}$ for $l = \{5,6,7,8,9,10\}$, respectively. The randomized scheme has smaller expected length (9.031) as well as better anonymity than the best deterministic choice $l = 10$.

Lastly, we consider the optimal probability distribution of $\mathcal{L}$ that maximizes the source-sink pair anonymity. From simulations, we observe that the performance depends on the probability distribution of $\mathcal{L}$. For example, probability distribution $P(l) = \{\frac{1}{32}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}\}$ in Fig. 4 outperforms all deterministic cases, but uniform distribution achieves smaller entropy than some of deterministic cases ($l = 9, 10$). The rule of thumb for a good probability distribution is that it should concentrate more on the longer subgraph length and the longer length should have larger probability than the shorter length (i.e., the probability monotonically increases with the length). The expected length should not be too smaller than

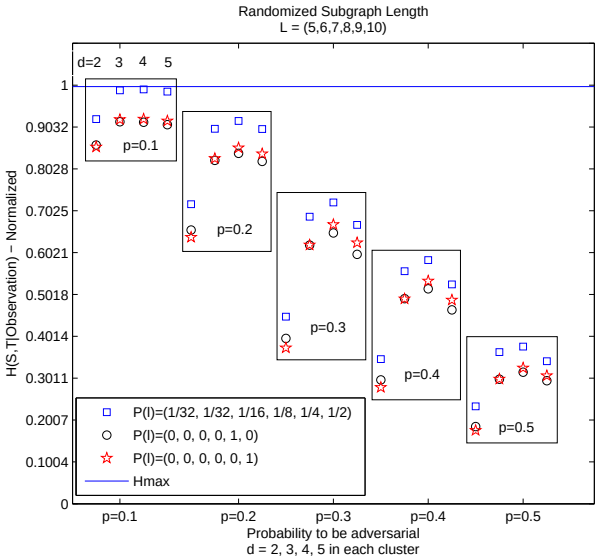

Fig. 4. Randomized subgraph: a source randomly chooses a subgraph length from a probability distribution, $\mathcal{L}$, and constructs a subgraph. Randomizing subgraph length simultaneously improves the anonymity (larger entropy) and the resource usage efficiency (smaller subgraph size) compared to the deterministic case.

the longest subgraph length. If the resource usage is sacrificed too much, then the anonymity will worsen as well. Once the criterion is satisfied, the difference in performances of different probability distributions is negligible (less than 1%) whereas the percentage difference between the randomized case and the deterministic case is over 10%.

IV. Data Transfer Phase

In this section, we assume availability of a subgraph setup scheme (either PKI-based as in [2], [3] or coding-based as in the previous section) that can be used to distribute coding/forwarding instructions to each node. We consider the use of network coding in the data transfer phase, assuming end-to-end encryption for data confidentiality. We propose a coding scheme involving column operations to address content correlation attacks. We study how the use of network coding, as well as the subgraph shape and connectivity, affect anonymity and congestion when the amount of anonymous traffic that can be carried by overlay links and nodes is limited. Such limits may be due to the use of traffic shaping to prevent traffic analysis, for instance, by carrying anonymous traffic in the payload of packets that mimic allowed traffic types. As before, there are N participating nodes, an unknown subset of which may be adversarial. The adversarial nodes collude to try to determine the identities of sources and sinks from their collective observations; we consider passive attacks only. Each source constructs a subgraph for anonymous communication to a sink node using a randomly chosen subset of available nodes.

A. Coding Scheme

Each unicast session transmits information over a rectangular subgraph. We consider two types of rectangular subgraphs parameterized by length l, width d and connectivity: random subgraph and parallel path subgraph. Network coding is carried out over the random subgraph as described below. The parallel path subgraph, which is used as comparison, employs conventional cryptographic transformations at each hop. This prevents non-connected adversarial nodes from deducing that they are in the same subgraph and effectively prevents them from colluding.

In the random subgraph, between two consecutive layers, each pair of nodes is connected with probability r, subject to the following constraints. 1) Each node's in-degree should be at least two $\{2, \ldots, d\}$ so that nontrivial network coding occurs; 2) Each node's out-degree should be at least one $\{1, \ldots, d\}$ to avoid a dead-end. To make the nodes neighboring the source have in-degree at least 2, the source layer contains at least 2 nodes. As in Section III, a source can use multiple IP addresses connected by secure channels.

For a random subgraph employing straightforward network coding, adversaries that are not connected may still be able to collude since the subspaces spanned by a sufficiently large collection of packets from disconnected sets of adversaries have a larger intersection if they are in the same session as compared to the case where they are in different sessions. Therefore, by correlating the subspaces of observed packets, adversaries may be able to infer that they are likely to be in the same session even if they are not connected. We propose a novel low-complexity technique to address this vulnerability.

In the network coding system, relaying nodes linearly combine the received packets with coefficients randomly chosen from a sufficiently large field. These linear combinations are row operations on the matrix whose rows correspond to the received packets [19]–[21], and are specified by a left multiplication matrix. In the proposed scheme, before (or after) the row operations, each relay node in layer i performs a column operation by right-multiplying with a matrix A_i. The matrix A_i for the i^{th} layer is specified by instructions from the source in the subgraph setup phase, and the matrices for different layers are distinct. The source node generates invertible matrices A_i $(i = 1, \ldots, l)$ independently at random, where the elements of a matrix are drawn from some distribution. In a practical implementation, instead of sending the whole matrix, the source sends random seeds to reduce overhead. Each random seed must produce an invertible random matrix. Using the random seed, each node can generate corresponding matrix for the column operation. To transmit message M, the source preprocesses the message by right multiplying with $A_l^{-1}A_{l-1}^{-1}\cdots A_2^{-1}A_1^{-1}$. Fig. 5 illustrates this scheme. Note that the message in the first layer (next to the source) $MA_l^{-1}A_{l-1}^{-1}\cdots A_2^{-1}$ and the message in the last layer (next to to the sink) M have different subspaces.

To see how this scheme prevents non-connected adversaries from deducing whether they are in the same subgraph, suppose that a trusted node v in layer k is connected to adversaries u_1, u_2, and w. v receives packets M_1 and M_2 from u_1 and u_2, respectively, and sends $M_v = c_1M_1A_k + c_2M_2A_k$ to w, where c_1 and c_2 are random coefficients from a

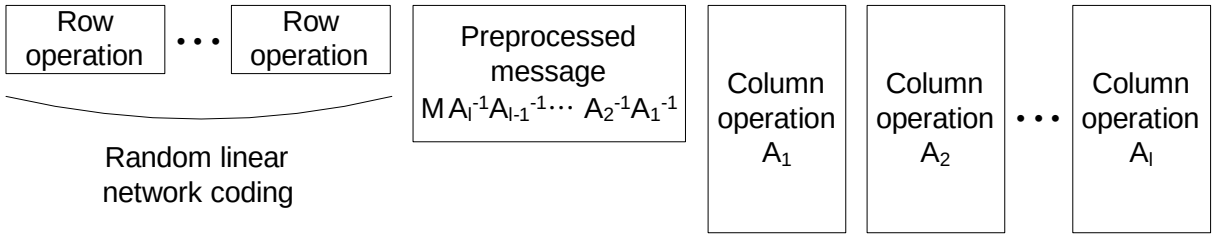

Fig. 5. In addition to the random linear network coding (row operation), each node performs column operation to the received packets that are preprocessed by the source in a systematic way.

sufficiently large field, and A_k is the column operation matrix for nodes in layer k. Without knowledge of A_k, the packets observed by u_1, u_2, and w appear unrelated. Adversaries can try all column operation matrices they possess, but if the probability p of a node being adversarial is reasonably small and the network contains a large number of nodes and sessions, this attack entails high overhead on the adversary's part and has low probability of success. Therefore, we assume that the adversary does not employ such an attack. Note that larger in-degree also increases the number of nodes that the adversary needs to control in order to be able to correlate packet contents. However, larger in-degree requires more source IP addresses. In the following, we consider the case of in-degree at least 2.

B. Anonymity and congestion performance

As discussed in the previous section, by employing cryptographic or matrix transformations at each hop, we assume that non-connected adversaries do not know that they are in the same subgraph and thus are unable to collude. The calculation of the entropy metric (1) is similar to the calculation in Section III-B except for the following. First, an adversarial cut does not reveal information about the downstream subgraph, unlike the earlier phase in which subgraph setup information is being sent using information theoretically secure coding; here we are concerned primarily with connected adversarial paths. Second, the calculation of $\mathcal{P}(a)$ reflects the subgraph connectivity structure in this phase. Details are given in [18, Chapter 3].

Traffic shaping to prevent traffic analysis limits traffic volume on overlay links/nodes. Also, unusually large node degree can be a fingerprint. Thus, we assume given constraints on link capacity and node out-degree. Link congestion is measured by the probability that the flow of a link exceeds its available capacity. To avoid exceeding the capacity of a link, the sending node can shift flow to other outgoing links if available. Also, a node can reject or drop connections in excess of the limit on out-degree, causing traffic to be shifted to other nodes. Further details are given in [18, Chapter 3]. We investigate congestion through Monte Carlo simulation experiments.

C. Simulation Results

For given problem parameters (network size $N = 100$, node out-degree limit $ODC = 20$, and the fraction of adversarial nodes $p = 0.2$), we analyze the anonymity and

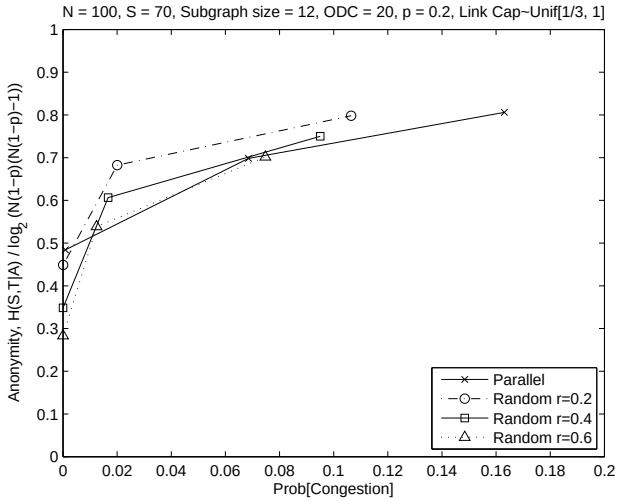

Fig. 6. Comparison for different subgraph shape, and connectivity. For a fair comparison, we set $S = 70$ for $ODC = 20$. In both cases, $(l, d, r) = (3, 4, 0.2)$ provides reasonably good anonymity and congestion at the same time.

congestion by controlling subgraph shape (l, d) and subgraph connectivity $r = \{0.2, 0.4, 0.6\}$. For a subgraph shape, we avoid the case of $d = 2$ that results in a degenerated case.[4] Due to the constraint of (in-degree) ≥ 2, any subgraph with $d = 2$ has a complete connection between the two adjacent layers (i.e., all nodes have in and out-degree 2). Then, adversaries neighboring sink can identify sink, since they have out-degree 1. Accordingly, the smallest width in our study is $d = 3$, for which the incoming flow to each node is $1/3$ in average. To make all nodes able to serve at least one session without exceeding link capacity, we consider the link capacities at least $1/3$. In the simulation, the link capacity is chosen independently at random from a uniform distribution in $[1/3,\ 1]$. In addition, we do not consider too high connectivity, which also results in a degenerated case—complete connection between the two adjacent layers. Therefore, we restrict the connectivity to $r \leq 0.6$.

Fig. 6 illustrates the tradeoff between anonymity and congestion probability for different out-degree limits, when the subgraph size is fixed at $ld = 12$ (candidate shapes:

[4]We also avoid trivial cases of $l = 1$ and $d = 1$. For $l = 1$, all adversaries are connected to source and sink. On the other hand, $d = 1$ results in trivial network coding.

TABLE I

THE NUMBER OF SESSIONS THAT CAN BE SERVED FOR GIVEN SUBGRAPH SHAPE (SIZE OF 18), CONNECTIVITY r, AND OUT-DEGREE LIMIT 20

$ODC = 20$	(l,d)		
r	$(2,9)$	$(3,6)$	$(6,3)$
0.2	61.17	58.82	58.66
0.4	47.02	49.83	56.14
0.6	36.28	40.58	52.88
Parallel	109.05	109.05	109.05

$(l,d) = \{(2,6),(3,4),(4,3)\}$). Note that the anonymity metric $H(S,T|\mathcal{A})$ is normalized by the number of all possible combinations for source-sink pair among all trusted nodes in the network, that is, $N(1-p)\,(N(1-p)-1)$ in average. A wide (large d) and short (small l) subgraph has good congestion performance but bad anonymity, since the flow is split more (less load on each link and therefore, good congestion) but it is easy to form an adversarial path spanning the whole length, revealing the source and sink identities (bad anonymity).

If out-degree limit is small, fewer sessions can be served. We observe that for $N = 100$, subgraph size 12, and $ODC = 20$, the smallest number of sessions (among candidates) that run out of available nodes is 73.1 in average when $(l,d,r) = (2,6,0.6)$. Therefore, for fair comparison (all candidates serve the same number of sessions without running out of nodes), we consider at most $S = 70$ in this case to compare the performances of different subgraph shapes and connectivity (Fig. 6). Table I shows the numbers of sessions that can be served by a subgraph size of 18 for 20 before running out of available nodes. In Fig. 7 we plot the envelope of Pareto optimal trade-off points between anonymity and congestion, by choosing optimal points from plots of anonymity versus congestion for each subgraph size as in Fig. 6.

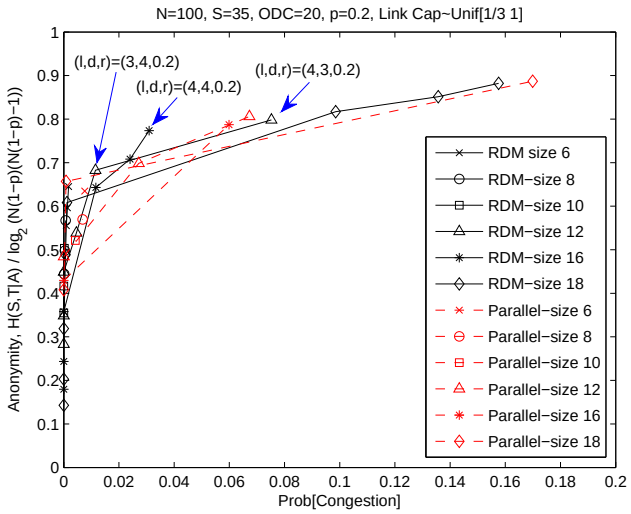

Fig. 7. To find the best subgraph shape and connectivity, we plot the "envelopes" from each subgraph size. For a fair comparison, we set $S = 35$ for $ODC = 20$. In this example, the best subgraph is either $(l,d,r) = (3,4,0.2)$, $(l,d,r) = (4,4,0.2)$, or $(l,d,r) = (4,3,0.2)$ depending on the relative importance between anonymity and congestion.

REFERENCES

[1] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[2] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE JSAC*, vol. 16, no. 4, pp. 482–494, 1998.

[3] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th Conference on USENIX Security Symposium*, p. 21, 2004.

[4] M. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *Proceedings of the 9th ACM CCS*, pp. 193–206, 2002.

[5] M. Rennhard and B. Plattner, "Introducing MorphMix: peer-to-peer based anonymous Internet usage with collusion detection," in *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pp. 91–102, 2002.

[6] A. Nambiar and M. Wright, "Salsa: a structured approach to large-scale anonymity," in *Proceedings of the 13th ACM CCS*, pp. 17–26, 2006.

[7] J. McLachlan, A. Tran, N. Hopper, and Y. Kim, "Scalable onion routing with torsk," in *Proceedings of the 16th ACM CCS*, pp. 590–599, 2009.

[8] M. Reiter and A. Rubin, "Crowds: Anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, 1998.

[9] A. Mislove, G. Oberoi, A. Post, C. Reis, P. Druschel, and D. Wallach, "Ap3: Cooperative, decentralized anonymous communication," in *Proceedings of the 11th Workshop on ACM SIGOPS European Workshop*, p. 30, 2004.

[10] S. Katti, D. Katabi, and K. Puchala, "Slicing the onion: Anonymous routing without pki," in *ACM HotNets*, 2005.

[11] Y. Fan, Y. Jiang, H. Zhu, J. Chen, and X. Shen, "Network coding based privacy preservation against traffic analysis in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 834–843, 2011.

[12] P. Gasti, A. Merlo, G. Ciaccio, and G. Chiola, "On the integrity of network coding-based anonymous p2p file sharing networks," in *Proceedings of the 9th IEEE International Symposium on Network Computing and Applications (NCA)*, pp. 192–197, 2010.

[13] J. Wang, J. Wang, C. Wu, K. Lu, and N. Gu, "Anonymous communication with network coding against traffic analysis attack," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1008–1016, 2011.

[14] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE JSAC*, vol. 20, no. 8, pp. 1489–1499, 2002.

[15] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, "Secure routing for structured peer-to-peer overlay networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 299–314, 2002.

[16] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies*, pp. 54–68, 2002.

[17] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies*, pp. 259–263, 2002.

[18] C. Chang, *Applications of Coding in Network Communications*. PhD thesis, California Institute of Technology, 2012.

[19] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[20] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[21] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.