Network Coding for DoS resistance

Max Suraev
Aalto university
msuraev@cc.hut.fi

Abstract

Denial of Service (DoS) attacks is a common problem in modern computer networks. There are many proposals trying to solve this problem in different cases, however no universal solution has appeared to date. Network Coding is a promising approach which could be used to increase network resistance to such attacks. This paper for the Seminar on Network Security reviews *network coding* techniques and their applicability to the mitigation of Denial of Service attacks.

1 Introduction

Since its introduction in 2000 [2] network coding has attracted much attention due to the remarkable simplicity of the main idea behind it: spreading information over packets traveling through the network by allowing forwarding nodes to perform recombination of information instead of plain forwarding. Network coding is used in many interesting research projects especially in wireless networks.

Denial of Service has been a part of day-to-day security challenges in the Internet for many years. Despite a lot of attention and research efforts, there is no universal solution for that problem. Moreover, even definitions and classifications varies significantly for different authors. For example [16] failed to distinguish between attack types and software tools implementing those attacks. This paper uses the classification based on [7] but includes some improvements.

Network coding itself has its own security challenges. However, in certain cases it could be a viable approach to mitigate effects of several classes of DoS attacks.

This paper is organized as follows: the classification of denial of service attacks is introduced in order to narrow down problem of applicability of network coding for attack mitigation. The short illustration of network coding is presented after that. The next section deals with security consideration related to network coding. Next potential scenarios for using network coding techniques against DoS attacks are described. The paper concludes with the discussion and the summary of the relations between network coding as a security measure and description of prospective research directions.

2 DoS attacks classification

There are several levels on which DoS attacks could be classified. The general overview of classification levels is illustrated in Fig. 1 and in general is similar to the one presented in [7]. Other variants of classifications considering attack impact and software used for attack are omitted for simplicity.

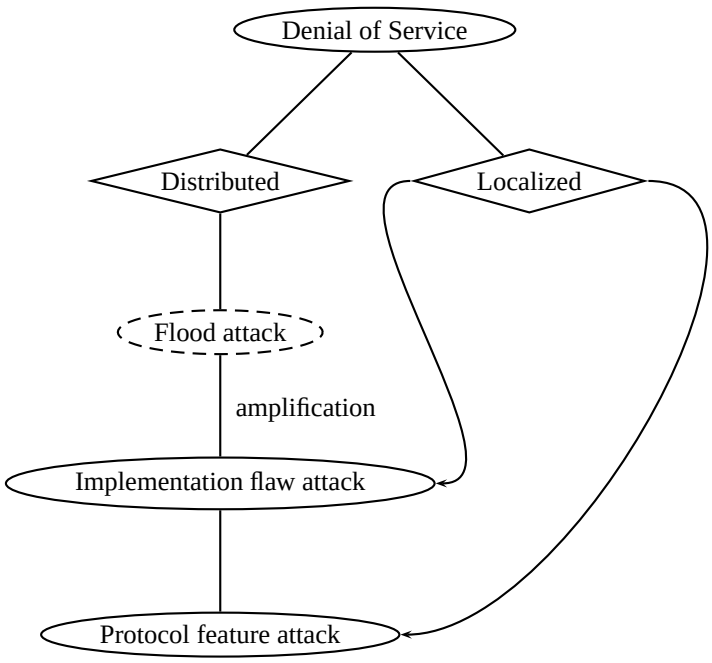

Figure 1: Compound DoS classification

First of all we can distinguish between distributed attacks involving many attacking agents and localized attacks. Another level of classification is a target level of attack: it could be against particular feature of a protocol (e. g. TCP SYN attack) or it could exploit some flaws in particular implementation of a protocol (e. g. Ping-of-Death attack against MS Windows).

Especially interesting is a case of Flood attack, when no particular flaw or protocol feature is targeted but resource exhaustion in general: regardless of how good protocol design and implementation were there are always some resource limits in practice (bandwidth, memory, CPU etc). This is a key property of DDoS (Distributed DoS) that's why there is no universal solution for DoS problem but rather a set of methods to mitigate its impact. One example is a method proposed in [13] which uses graphical tests to distinguish between regular human users and attacking bots trying to mimic user's behavior.

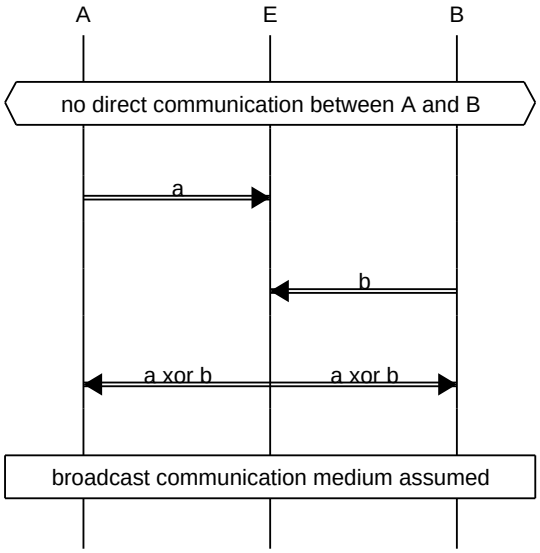

Figure 2: Simple Network Coding primer

Of course resource exhaustion is a two-fold problem: attacker has to deal with it too, even in DDoS case. That's why there is an ongoing research on attack amplification methods: the ways to force an attacked system to use order of magnitude more resources than attacker.

3 Network Coding

In contrast with modern ip networks comprising the Internet, Network coding allows intermediate nodes to recombine incoming packets to produce set of outgoing packets. The classical example is shown in Fig. 2. Using **xor** function for network coding we eliminate the need to transfer both packets a and b from intermediate point which allow us to use available bandwidth more effectively than traditional methods.

In practice, linear network coding is used for implementation: outgoing packets are computed as a linear combination of incoming packets. In this case incoming packets $X^1..X^n$ are viewed as a vectors over finite field F_{2^s}. Outgoing packets are computed as $Y_k = \sum_{i=1}^{n} g_i X_k^i$ where k is a position in appropriate vector and g_i are encoding coefficients. Each network node chooses its own encoding vector $g = \{g_i\}$ by randomly selecting encoding coefficients. The encoding vector is transmitted alongside with information vector (result of linear combination) to make decoding possible.

Decoding performed as follows: for received set of $\{g^j, Y^j\}$ system of linear equations $Y^j = \sum_{j=1}^{m} g_i^j X^i$ is solved with regards to X^i. Note: $m > n$ because some of the equations might be linearly dependent.

Linear network coding allows efficient implementation due to well-understood and optimized algorithms: Gaussian elimination for removing linearly dependent equations during decoding, polynomial arithmetics for multiplication and Euclid algorithm for division over F_{2^s}. For smaller fields ($s = 8$ for example) more efficient way via logarithm table lookup (similar to AES implementation) is available [8]. More formal and complete algebraic description of network coding is available at [15].

4 Security of Network Coding

The basic form of Network coding suffers from severe security flaw: it is very easy for attacker controlling one of the intermediate nodes to distribute corrupted packets over network. This could be seen as an example of traffic amplification mentioned in Sec. 2 - it is called *pollution attack*. The biggest advantage of network coding (effective information spreading) becomes its biggest security weakness: once one of the intermediate nodes controlled by attacker introduce single corrupted packet into data flow it is encoded by further intermediate nodes. It allows corruption to spread through entire network occupying bandwidth, CPU and other resources on multiple nodes without additional efforts from attacker. Data corruption introduced by pollution attack could be easily detected by receiving node but this will not help to recover wasted resources. That's why packets which are part of pollution attack should be detected and eliminated as close to source as possible. In order to do so we have to use security functions with security properties which are preserved by network coding scheme: this class of functions called *homomorphic*. There are several schemes using homomorphic hash functions [10], homomorphic signature scheme [4, 21] and homomorphic MAC (Message Authentication Code) proposed to prevent pollution attack.

However securing network coding is more difficult than it might seem for some researchers. For example schemes proposed in [5, 12] focuses only on protection against insecure links and becomes completely broken when attacker controls even single network node. This makes it impractical for securing wireless networks and in many other scenarios as well.

Scheme introduced in [21] is completely broken despite security proof given by authors: signature verification will always fail due to incorrect assumption[1] made by authors during scheme calculation.

Secure hashing scheme introduced in [10] is impractical because it require out-of-band secure channel for hash distribution. Comprehensive overview of attacks on various schemes (including [21], [12] and [17]) trying to secure network coding against pollution attack could be found in [20].

For wireless environments RIPPLE scheme [17] proposed to secure network coding but it require frequent periodical broadcasts of significant amounts of information (checksum packets) which is not suitable in wired p2p networks. Also it unable to prevent pollution attack when attacker control 2 or more nodes and have fast out-of-band communication channel between them [20].

Extension for signature scheme proposed in [4] for multi-source scenario reviewed in general in [1] and for more specific case in [6]. Homomorphic MAC proposed in [20] uses symmetric cryptography and delayed key release similar to TESLA [18] protocol. This estimated to be more computationally effective for practical implementations than [4].

[1] As noted in [9] equation (12) assume that $(A^b \mod p)^d \mod r = (A \mod p)^{bd} \mod r$ for a given A, b, d, prime p and RSA composite r. This obviously does not hold, for example: $A = 5, p = 7, q = 11, b = 3, d = 4$.

For **xor**-based network coding security scheme utilizing MACs for overlapping parts of the message proposed in [22]. It relies on symmetric cryptography and might not be generalized for random linear network coding.

Table 1: Summary of existing security proposals for network coding

Scheme	Comment
Cai et al. [5]	unrealistic threat model
Jaggi et al. [12]	
Yu et al. [21]	miscalculation of signature scheme
Gkantsidis et al. [10]	impractical deployment requirements
Li et al. [17]	unscalable threat model
Boneh et al. [4]	might be computationally expensive in some scenarios
Gennaro et al. [20]	might be inapplicable for multi-session environment
Yu et al. [22]	might be inapplicable for random linear network coding

There are no know attacks on [20], [22] and [4] schemes as of time of writing. Summary on various reviewed security schemes shown in Tab. 1. However all of them introduce overhead to network coding both in terms of processing capabilities of nodes and additional bandwidth. Simulations were performed to show practicalities of those schemes but large-scale implementations and tests are yet to be seen.

5 Applying network coding

Network coding could be especially useful in some scenarios. In this part cases for both wired and wireless network are examined with regards to applicability of network coding for DoS mitigation.

5.1 Wireless networks

Existing implementation of network coding for wireless networks promise significant improvements. The best example would be COPE project [14] which was implemented and tested in real wireless network environment. It demonstrated significant improvement comparing to existing 802.11 networks (from a few percent to several folds depending on traffic pattern). However project resistance to jamming (wireless DoS) is not examined. Combining COPE encoding with MAC proposed in [20] might help to secure at least against jamming employing pollution attack. This examined in more details in Sec. 5.1.2.

5.1.1 COPE

Wireless networks are naturally suit for implementing network coding because of broadcast nature of radio channels. Most wireless protocols today abstract this fact and use protocols from wired networks treating radio medium as a set of point-to-point links. Approach chosen by COPE project essentially boils down to example shown in Fig. 2. Of course implementation details are much more complicated: things like changes in radio environment and traffic patterns, client mobility are needs to be taken into account.

Main techniques employed by COPE are:

- **Opportunistic Listening:**

  Every node listen to all packets in promiscuous mode and store received packets for short period of time. Stored packets announced to other nodes via reception reports (either annotations attached to regular traffic packets or special control packets if no outgoing traffic present at the moment).

- **Opportunistic Coding:**

  Each node tries to maximize amount of original packets transmitted with encoded packets yet ensuring that each receiver have enough information to decode packets.

  This can be formulated as a following rule [14]:

  To transmit n packets, $p_1, \ldots, p_n$, to n next-hops, $r_1, \ldots, r_n$, a node can **xor** the n packets together only if each next-hop r_i has all $n-1$ packets p_j for $j = i$.

  When choosing which packets to transmit mode pick maximum n which satisfies above rule.[2]

- **Learning Neighbor State:**

  In addition to reception reports (which might be lost in congested network) each node utilize routing protocol metrics (for example ETX metric for OLSRv2 routing protocol) to guess probability that certain receiving node have particular packet. Incorrect guess will lead to undecodable packet for some of the receivers which equals to packet drop for upper layers in the protocol stack and might lead to packet retransmission.

Example of COPE operation is illustrated by Fig. 3. Router[3] in the middle have all the packets currently in transit and its signal is visible to all other nodes on the picture. Wireless nodes have different packet pools (shown in curly brackets) and expect various packets from router (shown as dotted arrows with labels). By broadcasting $g \oplus b \oplus r$ router is able to deliver 3 out of 4 packets expected by 3 nodes which is significant gain compared to conventional 802.11 networks:

$$\begin{cases} D : p = & g \oplus b \oplus r \\ A : r = & p \oplus b \oplus g \\ B : g = & p \oplus b \oplus r \\ C : b = & p \oplus r \oplus g \end{cases} \quad (1)$$

From practical point of view network coding have remarkable property: the more traffic in the network - the more coding options each COPE node have thus the more gain

[2] In general choosing optimal linear network coding is NP-hard [15] so COPE relies on set of heuristics for that.

[3] Router is just the node transmitting packets for other nodes in wireless mesh network. For example node **C** could serve as a router for nodes **B** and **D** as well.

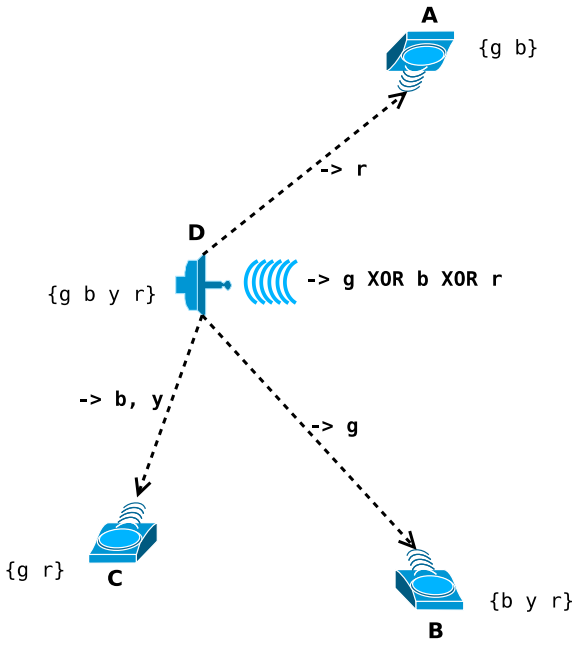

Figure 3: COPE packet exchange example

is obtained compared to conventional 802.11 network. Note: COPE uses inter-session linear network coding and is implemented between IP and MAC[4] layers of 802.11 protocol stack.

5.1.2 Secure homomorphic MAC

Protocol described in [20] assume loose synchronization of clocks on network nodes and following time boundaries: $[t_0 + r\delta, t_0 + (r+1)\delta)$ for $0 \leq r \leq T$ denotes *time* r where T is a maximum distance between nodes in the network (in number of hops) and δ is maximum time needed to deliver packet to next hop (including processing time).

- **Session key setup.**

  For each session id the source node choses random seed s and computes

  $$b^r = H_1^r(s, id) \in F_p,\ a^r = H_2^r(j, b^r) \in Fp \qquad (2)$$

  where H_1 and H_2 are pseudo-random functions and $H_1^r() = H_1(H_1(\ldots H_1()))$ for $r = 1 \ldots T+1$ and $j = 1 \ldots n+T$. Similar to Sec. 3 we assume $a^r = (a_1^r \ldots a_n^r)$.

  At the beginning of the session source signs (using traditional asymmetric cryptography) and broadcast b^{T+1} to all nodes. At time $\frac{r(r+1)}{2}$ source node also broadcast b^{T+1-r}(no signature required).

- **MAC generation.** Source node generates T MAC tags for the message $M \in F_p^n$:

  $$MAC_1(M) = a^1 M$$
  $$MAC_r(M) = a^r M + \sum_{j=1}^{r-1} a_{n+j}^r MAC_j(M) \qquad (3)$$

- **MAC Combination.** For session id intermediate node located r hops away from the source receive k tuples $(id, v_i, MAC_1(v_i) \ldots MAC_{T-r+1}(v_i))$ where $i = 1 \ldots k$. For each outgoing interface node chooses random $\alpha_1 \ldots \alpha_k \in F_p$ and send tuple $(id, v, MAC_1(v) \ldots MAC_{T-r+1}(v))$ where $v = \alpha_1 v_1 + \cdots + \alpha_k v_k$ and MAC tags computed as

  $$MAC_u(v) = \sum_{j=1}^{k} \alpha_j MAC_u(v_j) \qquad (4)$$

- **Message Transmission and MAC Verification.** Tuples received by intermediate node with distance r to source (approximate time is $\frac{r(r+1)}{2}$) are buffered until b^{T-r+1} is received (approximate time is $r + \frac{r(r+1)}{2}$). Correctness of received value verified as $H_1^r(b^{T-r+1}) = b^{T+1}$. After that $a_0^{T-r+1} \ldots a_n^{T-r+1}$ computed as in Eqn. 2. For each buffered tuple checked weather the following equation holds:

  $$MAC'_{T-r+1}(v) = a^{T-r+1} v + \sum_{j=1}^{T-r} a_{n+j}^{T-r+1} MAC'_j(v) \qquad (5)$$

  After packets who failed this check are discarded node continue with MAC combination process shown above.

More details and security proofs could be obtained in [20]. Note: this protocol implies notion of sessions, in contrast COPE uses inter-session coding semantics - this makes it impossible to use straightforward combination of those protocols. In order to adopt this protocol for COPE environment some changes are required to protocol definition of session and additional research on how those changes might affect security properties. In contrast protocol proposed in [22] seems to perfectly fit into COPE environment however additional evaluation is required to verify that its security.

5.2 Web attacks

There are two ways to utilize network coding properties to mitigate impact of DoS attacks on web sites. One is to introduce caching front-end infrastructure and use network coding for communication between front-end and back-end. Another way is to use network coding for in-band communication with servers under attack.

5.2.1 Caching infrastructure

One of the common ways to deal with DDoS attack for web sites is by adding more resources to handle client traffic. This could be done using 2-tier architecture by splitting existing system into set of back-end servers responsible for con-

[4]Here MAC means Media Access Control. Throughout this paper MAC means Message Authentication Code unless specified otherwise.

tent generation and front-end servers responsible for content caching and handling of client connections.

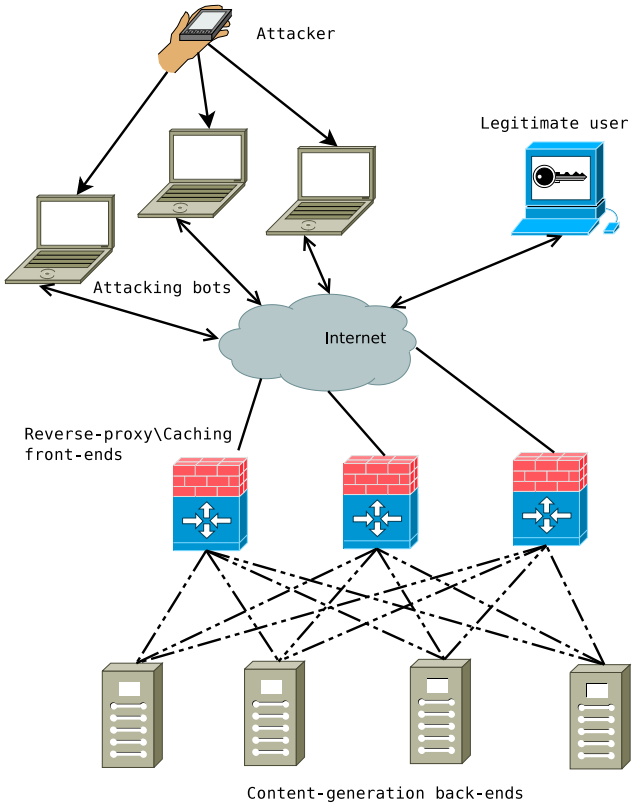

Figure 4: 2-tier infrastructure

In this case network coding could be used in communication between front-end and back-end (shown as double-dotted lines in Fig. 4) to more effectively utilize available bandwidth and leverage data from different client sessions for encoding.

5.2.2 In-band communication

Introducing additional resources might not be a feasible solution for many scenarios. In this case maintenance personnel have to use in-band communication in order to get access to server under DoS using same medium as both legitimate clients and attackers. Using conventional protocols based on TCP might be difficult or impossible due to heavy packet loss.

Network coding effectively eliminate need for packet retransmission in this case because server only have to get any n packets in order to decode n original packets. This property could be used to establish connection with the server despite unreliability of communication introduced by DoS.

5.3 Peer-to-peer networks

The have been several attempts to introduce network coding into peer-to-peer networks to both increase efficiency and DoS resilience. One of the ideas is to combine network coding with widely used BitTorrent protocol. There have been several short papers published on this topic however none of the attempts resulted in a program mature enough to be published as open source project. For example BitCod [3] is only a proposal of a set of simple heuristics which was never verified using network simulation let alone actual implementation. Security aspects of network coding were also omitted by authors.

Another recent paper [19] has shown that benefits provided by network coding for BitTorrent peer-to-peer network are negligible. Comparison between regular torrent clients and network coding enabled variant (named NCTorrent) was performed using large database of real download patterns collected in the Internet. In case of small block sizes NCTorrent performed slightly better but in other scenarios (small upload bandwidth for example) it performed worse.

Authors of [11] tried to build file distribution network using network coding and compare its performance to BitTorrent. However they used simulation instead of real programs for comparison and failed to implement proper specification of BitTorrent protocol so the results of this comparison are lacking relevance and practical significance.

6 Discussion

Network coding cannot be viewed as the universal solution for the problem of DoS attacks. However, in some scenarios it could lower the effects of such attacks either by more effective utilization of communication medium between server and clients or by allowing more reliable communication for server administrator.

One of the crucial points is a practical implementation and comparison with existing implementations. For example authors of COPE project made accurate and detailed performance comparison of their network coding implementation with existing mesh networks based on 802.11 stack using network coding for packet forwarding and transparently integrating with routing and upper layers. In contrast, applications utilizing network coding for peer-to-peer networks are yet to be seen.

7 Conclusion

Security shortcomings of network coding were reviewed alongside proposed enhancements to overcome them. Using a given classification of DoS attacks, several cases were examined in more details to show how network coding might be used as a mitigation measure.

Wireless networks provide great possibilities for using network coding due to the broadcast nature of radio communication. Practical implementation of network coding in COPE architecture were examined. The homomorphic MAC protocol was considered as a potential security measure against DoS attacks against COPE. However, reviewed protocol requires additional changes due to different treatment of session in the COPE architecture.

Two potential use cases for network coding were proposed for wired networks with web servers as an example target for DoS attack. First allows web servers to tolerate more intense DDoS attacks by more efficient bandwidth utilization between web servers and caching servers. Second allows direct access to the web server via the link congested by attack

traffic by using network coding to eliminate need for packet retransmission.

Using network coding is a two-fold idea: on the one hand it might help to decrease the impact of DoS attacks, on the other hand it requires special authentication protocols in order to prevent potential DoS attack targeting information dissipation properties of network coding itself.

7.1 Future work

Rigorous testing with real-life traffic patterns and workload scenarios required to demonstrate feasibility of proposed methods. Especially interesting is a combination of secure homomorphic authentication schemes with COPE wireless mesh network architecture.

References

[1] S. Agrawal, D. Boneh, X. Boyen, and D. M. Freeman. Preventing pollution attacks in multi-source network coding. Cryptology ePrint Archive, Report 2010/183, 2010. `http://eprint.iacr.org/`.

[2] R. Ahlswede, N. Cai, S. yen Robert Li, R. W. Yeung, S. Member, and S. Member. Network information flow. *IEEE Transactions on Information Theory*, 46:1204–1216, 2000.

[3] D. Bickson and R. Borer. The bitcod client: A bittorrent clone using network coding. In *Proceedings of the Seventh IEEE International Conference on Peer-to-Peer Computing*, P2P '07, pages 231–232, Washington, DC, USA, 2007. IEEE Computer Society.

[4] D. Boneh, D. Freeman, J. Katz, and B. Waters. Signing a linear subspace: Signature schemes for network coding. In *Irvine: Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography*, pages 68–87, Berlin, Heidelberg, 2009. Springer-Verlag.

[5] N. Cai and R. W. Yeung. Secure network coding. In *Proc. IEEE ISIT'02*, June 2002.

[6] L. Czap and I. Vajda. Signatures for multi-source network coding. Cryptology ePrint Archive, Report 2010/328, 2010. `http://eprint.iacr.org/`.

[7] C. Douligeris and A. Mitrokotsa. Ddos attacks and defense mechanisms: classification and state-of-the-art. *Comput. Netw.*, 44(5):643–666, 2004.

[8] C. Fragouli, J.-Y. Le Boudec, and J. Widmer. Network coding: an instant primer. *SIGCOMM Comput. Commun. Rev.*, 36(1):63–68, 2006.

[9] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin. Secure network coding over the integers. Cryptology ePrint Archive, Report 2009/569, 2009. `http://eprint.iacr.org/`.

[10] C. Gkantsidis and P. Rodriguez. Cooperative security for network coding file distribution. In *INFOCOM*, 2006.

[11] C. Gkantsidis and P. R. Rodriguez. Network coding for large scale content distribution. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2235–2245. IEEE, March 2005.

[12] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Médard, and M. Effros. Resilient network coding in the presence of byzantine adversaries. *IEEE Transactions on Information Theory*, 54(6):2596–2603, 2008.

[13] S. Kandula, D. Katabi, M. Jacob, and A. Berger. Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds. In *In 2nd Symposium on Networked Systems Design and Implementation (NSDI*, 2005.

[14] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: practical wireless network coding. *IEEE/ACM Trans. Netw.*, 16(3):497–510, 2008.

[15] R. Koetter, M. Médard, and S. Member. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11:782–795, 2003.

[16] F. Lau and S. H. Rubin. Distributed denial of service attacks. In *In IEEE International Conference on Systems, Man, and Cybernetics*, pages 2275–2280, 2000.

[17] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen. Ripple authentication for network coding. In *INFOCOM'10: Proceedings of the 29th conference on Information communications*, pages 2258–2266, Piscataway, NJ, USA, 2010. IEEE Press.

[18] A. Perrig, J. D. Tygar, D. Song, and R. Canetti. Efficient authentication and signing of multicast streams over lossy channels. In *SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy*, page 56, Washington, DC, USA, 2000. IEEE Computer Society.

[19] S. Wang, J. Zhao, and X. Wang. Is network coding helpful for bittorrent: From a practitioner's perspective. In *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pages 1 –6, 2010.

[20] Y. Wang. Insecure "provably secure network coding" and homomorphic authentication schemes for network coding. Cryptology ePrint Archive, Report 2010/060, 2010. `http://eprint.iacr.org/`.

[21] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan. An efficient signature-based scheme for securing network coding against pollution attacks. In *INFOCOM'08*, pages 1409–1417, 2008.

[22] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan. An efficient scheme for securing xor network coding against pollution attacks. In *INFOCOM*, pages 406–414. IEEE, 2009.