

## Chapter 2

# Security Threats in Network Coding

While network coding can help improve network performance, it also introduces new security and privacy issues. For example, network coding can make data transmission more vulnerable to “pollution attacks”: A single illegal packet can end up polluting a bunch of good ones through the process of intermediate coding, causing receivers unable to decode properly. Besides data integrity, data confidentiality and user privacy also become quite different in the new environment of network coding. This makes existing schemes like digital signatures, encryption algorithms, and anonymity schemes either infeasible or inefficient. This chapter will introduce several attacks on network coding and the corresponding countermeasures.

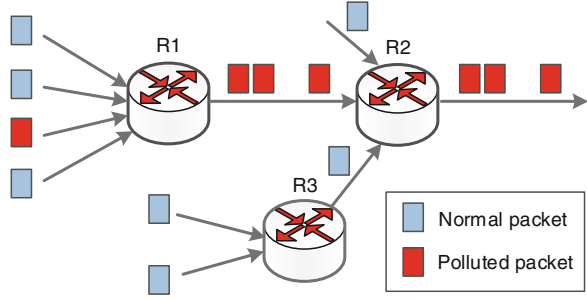
### 2.1 Pollution Attack

Among all the threats of network coding considered so far, pollution attacks are perhaps the most concerned ones. Figure 2.1 gives an example of pollution attack in network coding. We can see that a single polluted packet can end up polluting many more good ones. To thwart this kind of attacks, many schemes have been proposed. We classify them into two categories: *information-theoretic schemes* and *cryptography-based schemes*.

#### 2.1.1 Information-Theoretic Schemes

Information-theoretic schemes mostly leverage error correction codes to add redundancy to the messages at source nodes. In this way, destination nodes can recover the original messages from the received packets (which may contain polluted packets) through decoding. Methods proposed in [9, 49, 53] can correct the transmission

**Fig. 2.1** Example of pollution attack in network coding



errors in network coding. They are not designed to detect pollution attack; they can only mitigate the attack to some extent. Ho et al. [21] propose an extension to random linear network coding to detect Byzantine modification (i.e., pollution attack), by adding hashes in the source messages. Jaggi et al. [23] propose three different polynomial-time source coding methods. Suppose the communication capacity is  $C$  and the bandwidth of the attack is  $z_o$ . Then, a secure throughput of  $C - z_o$  or  $C - 2z_o$  can be achieved using these codes.

Kotter et al. [28] model the error correction of network coding as the problem of transmitting a *vector space* through an operator channel. The authors observe that the only invariant of error-free random linear network coding is the linear space spanned by packets. By checking whether the vector space generated by the received packets is the same as that generated by the source packets, the receiver can detect whether there is a transmission error or malicious modification. Formally, suppose the source transmits a linear subspace  $V$ , each of whose basis corresponds to a source packet. Since  $V$  is closed under linear combinations, it will remain the same after network coding. Similarly, the destination also constructs a linear subspace  $U$  from all received packets. If  $\text{Rank}(V \cap U)$  is sufficiently large, the transmission can be said to be error-free. Based on this idea, the authors introduced a new metric termed *rank distance* and proposed an optimal coding scheme similar to Reed-Solomon codes [47]. Different from [23], this coding scheme can operate on any finite field and has no restriction on packet size.

### 2.1.2 Cryptography-Based Schemes

To actively prevent pollution from propagating among intermediate nodes, several *cryptography-based schemes* have been proposed. They can be further grouped into two classes: *public key cryptographic approaches* and *symmetric key cryptographic approaches*.

**Public-Key Cryptographic Approaches** In the innovative work of Krohn et al. [30], *homomorphic hash function* is proposed to enable on-the-fly verification for erasure codes. As the verification process requires nodes to compute expensive

homomorphic hash functions, the technique of *batched verification* is employed. Gkantsidis et al. [20] extend this scheme to network coding-based P2P networks and further reduce the computation overhead by enabling cooperative verification among peers. One common limitation of these two schemes is that the hash values of the whole file should be computed at the source and delivered to downstream nodes in advance.

To address the above problem, Yu et al. [50] propose a *homomorphic signature* scheme on basis of homomorphic hash function and RSA cryptosystem. In their scheme, each packet carries an RSA-encrypted homomorphic hash, which functions as its homomorphic signature. A recent work [52], however, shows that this signature scheme is only conditionally valid and may be vulnerable to trivial no-message attacks. Charles et al. [10] introduce a secure homomorphic signature based on Weil pairing over elliptic curves [6], which is even more expensive than homomorphic hash functions.

From quite a different perspective, Zhao et al. [54] propose a signature scheme in which relay nodes check the integrity of packets by verifying whether they belong to the subspace of source packets. Boneh et al. [7] introduce another signature scheme similar to [54], but use signatures of a smaller size. However, both [54] and [7] still require extra secure channels to pre-distribute signatures, just like [20, 30]. To sum up the above public key cryptographic approaches, they are not computationally efficient due to the expensive operations of homomorphic hashing or signatures.

**Symmetric Key Cryptographic Approaches** In the scheme proposed by Yu et al. [51], the source attaches to each packet multiple MACs, each of which authenticates part of the packet. These MACs are encrypted using different keys at the source, and relay nodes can cooperatively check different MACs using their respectively shared keys. Agrawal et al. [2] propose a similar MAC-based scheme, which differs from [51] in that each MAC authenticates the whole packet. However, this scheme is unfortunately vulnerable to tag pollution, which can also degrade the system performance.

Toward this problem, Li et al. [33] propose RIPPLE, a time-based authentication protocol based on TESLA [38]. The key idea of RIPPLE is time asymmetry: the source utilizes a stream of keys to generate multiple tags for each packet, but does not pre-distribute these keys. The keys are distributed only when nodes need to verify a packet. In this way, an adversary does not know the keys being used in prior and thus cannot forge valid tags. Similarly, Dong et al. [16] propose another TESLA-based method named DART. In DART, the source node continuously generates random MACs after it has sent packets. Other nodes only use the MACs that are generated “after” the time when they received the packets for verification. A common problem with RIPPLE and DART is that they require global synchronization among all nodes, which is not easy to be realized in distributed settings. In addition, since nodes need to wait for the keys or tags, they will incur an additional transmission latency.

Kehdi et al. [27] propose another symmetric key-based scheme, which utilizes *null keys* for verification. A null key is just a vector from the null space of the matrix

formed by the source packets; legitimate packets are supposed to map null keys to zero. This scheme may incur a high bandwidth overhead since it injects into the network with multiple null keys for each generation. In sum, the above symmetric key cryptographic approaches are quite efficient in computation, but they must have carefully manage the keys and incur a relatively large bandwidth overhead.

They are many other schemes for defending against pollution attacks [25, 31, 32], and we will not cover them for limited space.

## 2.2 Eavesdropping Attack

There are several works focusing on secure network coding against eavesdropping attacks. We broadly group them into the following three classes according to the security level that they aim to achieve:

### 2.2.1 Shannon Security

The first category of works attempt to provide *Shannon security* by designing secure linear codes for network coding. Cai et al. [8] first identify the wiretap adversary who can monitor a limited number of links and present an approach to transform any linear network code to be secure using secret sharing. More specifically, suppose the max-flow capacity of the network is  $n$ , and the adversary can monitor at most  $k$  edges; then the sender should send  $n - k$  message symbols with another  $k$  random symbols, so that no information will be leaked to the adversary. However, Cai's scheme is not very efficient as it requires a large field size and takes a large number of steps.

This problem is treated by Feldman et al. [19], who generalize the method used in [8] and find a trade-off between the multicast capacity and the field size. In this way, the field size can be significantly reduced by giving up a small amount of capacity. An interesting observation made in [19] is that making network coding secure is equivalent to finding codes with some distance properties. Rouayheb et al. [17] formalize the problem as a generalization of Ozarow–Wyner wiretap channel II model [37] and propose a secure scheme by implementing coset coding at the source. They show that this coding scheme can be implemented without affecting the underlying network coding architecture. This model is further studied by Silva et al. in [44], where another source coding scheme, named maximum rank distance (MRD), is designed to replace the codes used in [17]. Both [17] and [44] stress that the designing of secure codes and the optimizing of network transport can be treated independently.

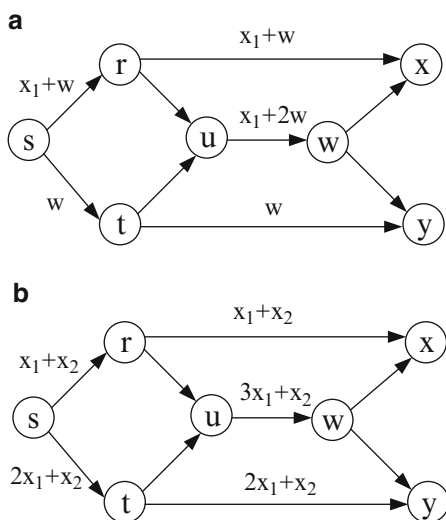
Different from the above approaches, Adeli et al. [1] propose a different secure network coding scheme with small overhead by utilizing hash functions. In their

scheme, each generation includes one uniformly distributed symbol as a secret, and the randomness of this symbol is scattered into other data symbols using a hash chain. This scheme is claimed able to provide complete security for linear network coding, regardless of the number of independent coding vectors acquired by the adversary. However, a more extensive proof is needed to justify this argument. To sum up the above schemes, they all assume that the eavesdroppers are limited in listening capability and unnecessarily sacrifice certain capacity to achieve Shannon security.

### 2.2.2 Weak Security

Bhattach et al. [4] introduce the concept of *weak security*, by which the system is said to be secure if the adversary cannot recover any “meaningful” information. To illustrate this concept, consider Fig. 2.2, for example. In Fig. 2.2a,  $x_1$  is the message to be transmitted, and  $w$  is a randomly generated key. In order to recover the source message  $x_1$ , the attacker should at least eavesdrop at two different links. Then, the transmission is Shannon secure given the adversary can only eavesdrop on one link. In Fig. 2.2b,  $x_1$  and  $x_2$  are the source messages to be transmitted. An adversary that eavesdrops on one link can still obtain some information (linear combinations of  $x_1$  and  $x_2$ ). In this sense, the transmission is not Shannon secure. However, the attacker cannot decode either  $x_1$  or  $x_2$  and thus cannot recover only “meaningful” information about them. Then, the transmission is said to be weakly secure. As seen in Fig. 2.2b, the source can transmit two messages simultaneously, meaning that weak security condition permits a higher transmission throughput compared

**Fig. 2.2** Comparison of Shannon security and weak security [4]. (a) Shannon secure transmission. (b) Weakly-secure transmission



to Shannon security. The authors also show that the weak security and multicast capacity can be simultaneously achieved, by performing linear transformations at the source. They also show that random linear network coding is inherently weakly secure with a high probability if coding is performed over a large finite field.

### 2.2.3 Computational Security

Lima et al. [34] consider the threats posed by “nice but curious” intermediate nodes and develop an algebraic security criterion to access the intrinsic security provided by network coding. They derive the relationship between field size and the security level and observe that the security is dependent on network topology. The algebraic security criterion is essentially weak security. Based on the weak security model, Wang et al. [46] design a polynomial-time deterministic code to secure linear network coding. They show that by using this scheme, optimal throughput for multiple streams between a single-source destination pair can be achieved.

By leveraging the intrinsic security of network coding, some cryptographic approaches have been proposed to secure network coding-based applications. One scheme is SPOC [35, 45], proposed by Vilela et al., in which the source encrypts/locks the GEV of each message after random linear coding and attaches another set of GEVs to enable standard network coding. Receivers can recover the source messages by following a decode-decrypt-decode procedure. This scheme is essentially an end-to-end cryptographic approach and is lightweight in computation. This scheme requires the source performs the following four steps before transmission: (1) prefix each message with the corresponding unit vector acting as coding vector; (2) linearly combine the prefixed messages using random coefficients; (3) perform symmetric encryption on the coding vector of each message; and (4) attach another coding vector as prefix for each message. After these four steps, the messages are sent, and intermediate nodes perform standard network coding on this packet. Sinks can recover the source messages by following a decode-decrypt-decode procedure.

Another scheme proposed by Fan et al. [18] is based on homomorphic encryption function (HEF) [3]. This scheme has the coding coefficients encrypted using HEF. Due to the homomorphic property of HEF, linearly combined operations can be directly performed on the encrypted coding coefficients. As a result, no extra coding coefficients are needed as by SPOC. As another difference from SPOC, Fan’s HEF-based scheme can achieve both content secrecy (i.e., confidentiality) and contextual secrecy (i.e., privacy) at the same time. However, both of these two schemes fail to fully exploit the mixing nature of network coding.

## 2.3 Entropy Attack

Entropy attack [24] is a new DoS attack that is specially targeted at network coding. In entropy attack, the adversary injects packets that are legal, meaning that these packets are linear combinations of source packets. However, these packets cannot increase the rank of the subspace generated by received packets and thus are not useful for receivers to decode. This attack gets its name since the entropy of the coded packets becomes low when the attack is launched. Entropy attack can cause a waste of bandwidth resource and reduce the probability of successful decoding at receivers.

Take Fig. 2.3 as an example. The source  $S$  tries to send some packets to the destination  $D$ . An adversarial node  $R_1$  residing in the network,  $R_2$  outside the network tries to launch entropy attacks. Suppose at each time slot  $t_i$ , the source  $S$  sends two coded packets  $y_{t_i}(e_1)$  and  $y_{t_i}(e_2)$  through two output links  $e_1$  and  $e_2$ , respectively.  $R_1$  continuously uses packets  $y_{t_i}(e_1)$  and  $y_{t_i}(e_2)$  to generate “non-innovative” packets  $y_{t_{3+}}(e_3) = ay_{t_i}(e_1) + by_{t_i}(e_2)$  and sends them to destination  $D$ . Similarly,  $R_2$  can also generate and inject “non-innovative” packets  $y_{t_{2+}}(e'_3) = a'y_{t_i}(e_1) + b'y_{t_i}(e_2)$  with intercepted packets. In this way, the destination  $D$  will not receive enough innovative packets and cannot decode the source packets.

In order to prevent the entropy attack, we can let nodes check the linear dependence of their received packets, and a new packet that is a linear combination of already received packet should be discarded. However, since the finite field size and packet size are relatively large, checking linear dependence will bring high computation load to nodes. To address this problem, Jiang et al. [24] propose a probabilistic algorithm that can detect and filter non-innovative packets fast. The authors show that when the finite field size is  $2^{256}$  and the packet length is 384, the computation overhead is only 4 % of the traditional linear dependence checking cost.

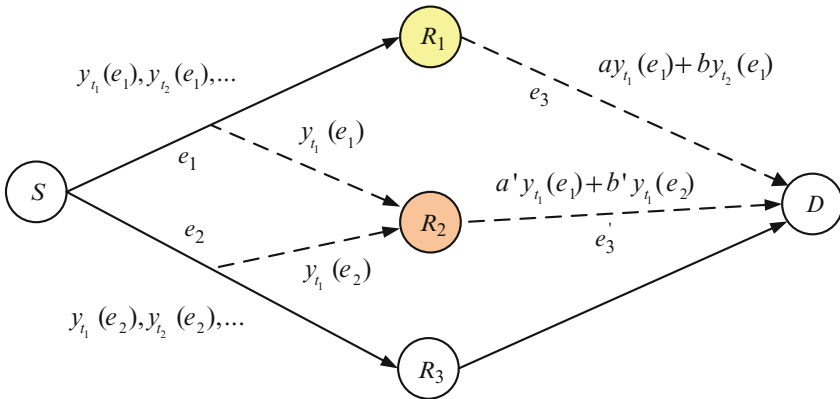


Fig. 2.3 Example of entropy attack in network coding [24]

## 2.4 Privacy Leakage

User privacy is a great concern in today's Internet. For wireline networks, there are two fundamental techniques, i.e., Mix-Net and Onion Routing. Here, Mix-Net refers to schemes that are based on Chaum's mix [11], e.g., Mixminion [12], Mixmaster [36] and MorphMix [41, 42]. The common feature of them is that they all employ techniques such as *shaping*, *reordering*, and *layered encryption* to eliminate the packet correlations at participating nodes. By layered encryption, the source should successively encrypt each packet with public keys of the nodes along the route. Then, each node peels one layer of encryption with its private key so that the packet finally arrives at the receiver as plaintext.

Onion Routing [39] refers to a family of anonymity protocols, which are also based on the technique of layered encryption, but is more computationally efficient than Mix-Net. In traditional Onion Routing, the source creates a layered structure named *onion*, by successively encrypting session keys for nodes along the route using their corresponding public keys. Then, each node along the route decrypts the onion with its private key to obtain the session key for it. After that, data packets are moved along the route just as in Mix-Nets, except that here packets are symmetrically encrypted by the source with the session keys previously distributed. The technique of Onion Routing is further developed in Tor [15], which ensures forward secrecy for Onion Routing using incremental path building, and in [26], which eliminates the need of PKI in Onion Routing using multi-path routing. Based on Tor, DiBenedetto et al. propose ANDaNA [14], which can provide anonymity for content centric networks [22].

As noted above, both Mix-Net and Onion Routing require relay nodes to perform encryptions/decryptions on packets, and these operations are in conflict with the packet-mixing operations required by network coding. Thus, neither Mix-Net nor Onion Routing can be applied to networks equipped with network coding.

Let us examine some typical anonymity techniques designed for wireless networks. Onion Ring [48] is an anonymity scheme proposed for wireless mesh networks. Unfortunately, since Onion Ring is based on Onion Routing, it cannot support network coding either. ANODR [29] provides an untraceable on-demand routing scheme which can protect user identities in multi-hop ad hoc networks. By using broadcast with tap-door information, ANODR supports distributed route discovery between two arbitrary nodes without revealing sender and/or receiver identities. WAR [5] is another anonymity scheme that exploits the broadcast nature of wireless networks. WAR differs from ANODR in that it has the initiating node select the transmission path, and uses cover traffic to thwart global eavesdropping. However, both ANODR and WAR still need to perform layered encryptions/decryptions on packets, just as in Mix-Net. Thus, they still cannot function when wireless network is upgraded to use network coding.

Different from the abovementioned schemes, Crowds [40] is an anonymity scheme designed for web transactions and is not based either in Mix-Net or Onion Routing. In Crowds, each sender will forward its web requests to a set of random



chosen members before the request reaches the web server. Thus, Crowds can incur a considerable delay and is not suitable for most applications with requirement of real-time communications. There are some variants of Crowds. For example, Hordes [43] reduces the transmission latency of Crowds using multicast routing, and D-Crowds [13] generalizes Crowds to a TTL-based deterministic forwarding scheme.

## References

1. Adeli, M., Liu, H.: Secure network coding with minimum overhead based on hash functions. *IEEE Commun. Lett.* **13**(12), 956–958 (2009)
2. Agrawal, S., Boneh, D.: Homomorphic macs: MAC-based integrity for network coding. In: *Proceedings of Applied Cryptography and Network Security*, pp. 292–305 (2009)
3. Benaloh, J.: Dense probabilistic encryption. In: *Proceedings of the Workshop on Selected Areas of Cryptography*, pp. 120–128 (1994)
4. Bhattad, K., Narayanan, K.R.: Weakly secure network coding. In: *Proceedings of International Symposium on Network Coding (NetCod)* (2005)
5. Blaze, M., Ioannidis, J., Keromytis, A.D., Malkin, T.G., Rubin, A.: Anonymity in wireless broadcast networks. *Int. J. Netw. Secur.* **8**(1), 37–51 (2009)
6. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: *Proceedings of Advances in Cryptology – ASIACRYPT*, pp. 514–532 (2001)
7. Boneh, D., Freeman, D., Katz, J., Waters, B.: Signing a linear subspace: signature schemes for network coding. In: *Public Key Cryptography (PKC)*, pp. 68–87. Springer, Heidelberg (2009)
8. Cai, N., Yeung, R.W.: Secure network coding. In: *Proceedings of International Symposium on Information Theory (ISIT)*, p. 323 (2002)
9. Cai, N., Yeung, R.W.: Network error correction, II: lower bounds. *Commun. Inf. Syst.* **6**(1), 37–54 (2006)
10. Charles, D., Jain, K., Lauter, K.: Signatures for network coding. *Int. J. Inf. Coding Theory* **1**(1), 3–14 (2009)
11. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM* **24**(2), 84–90 (1981)
12. Danezis, G., Dingleline, R., Mathewson, N.: Mixminion: design of a type III anonymous remailer protocol. In: *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, pp. 2–15 (2003)
13. Danezis, G., Diaz, C., Käsper, E., Troncoso, C.: The wisdom of crowds: attacks and optimal constructions. In: *ESORICS* (2009)
14. DiBenedetto, S., Gasti, P., Tsudik, G., Uzun, E.: Andana: anonymous named data networking application. In: *Proceedings of the Network and Distributed System Security Symposium (NDSS)* (2004)
15. Dingleline, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: *Proceedings of the 13th USENIX Security Symposium* (2004)
16. Dong, J., Curtmola, R., Nita-Rotaru, C.: Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks. In: *Proceedings of the Second ACM Conference on Wireless Network Security*, pp. 111–122 (2009)
17. El Rouayheb, S.Y., Soljanin, E.: On wiretap networks II. In: *Proceedings of International Symposium on Information Theory (ISIT)*, pp. 551–555 (2007)
18. Fan, Y., Jiang, Y., Zhu, H., Shen, X.: An efficient privacy-preserving scheme against traffic analysis attacks in network coding. In: *Proceedings of IEEE INFOCOM*, pp. 2213–2221 (2009)

19. Feldman, J., Malkin, T., Stein, C., Servidio, R.: On the capacity of secure network coding. In: Proceedings of the 42rd Allerton Conference on Communication, Control, and Computing (2004)
20. Gkantsidis, C., Rodriguez, P.: Cooperative security for network coding file distribution. In: Proceedings of IEEE INFOCOM, vol. 6, pp. 1–13 (2006)
21. Ho, T., Leong, B., Koetter, R., Médard, M., Effros, M., Karger, D.R.: Byzantine modification detection in multicast networks with random network coding. *IEEE Trans. Inf. Theory* **54**(6), 2798–2803 (2008)
22. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2009), pp. 1–12 (2009)
23. Jaggi, S., Langberg, M., Katti, S., Ho, T., Katabi, D., Médard, M.: Resilient network coding in the presence of byzantine adversaries. In: Proceedings of IEEE INFOCOM, pp. 616–624 (2007)
24. Jiang, Y., Fan, Y., Shen, X., Lin, C.: A self-adaptive probabilistic packet filtering scheme against entropy attacks in network coding. *Comput. Netw.* **53**(18), 3089–3101 (2009)
25. Jiang, Y., Zhu, H., Shi, M., Shen, X.S., Lin, C.: An efficient dynamic-identity based signature scheme for secure network coding. *Comput. Netw.* **54**(1), 28–40 (2010)
26. Katti, S., Cohen, J., Katabi, D.: Information slicing: anonymity using unreliable overlays. In: Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation (NSDI) (2007)
27. Kehdi, E., Li, B.: Null keys: Limiting malicious attacks via null space properties of network coding. In: Proceedings of IEEE INFOCOM, pp. 1224–1232 (2009)
28. Koetter, R., Kschischang, F.R.: Coding for errors and erasures in random network coding. *IEEE Trans. Inf. Theory* **54**(8), 3579–3591 (2008)
29. Kong, J., Hong, X.: Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing, pp. 291–302 (2003)
30. Krohn, M.N., Freedman, M.J., Mazieres, D.: On-the-fly verification of rateless erasure codes for efficient content distribution. In: Proceedings of IEEE Symposium on Security and Privacy (S&P), pp. 226–240 (2004)
31. Le, A., Markopoulou, A.: Tesla-based defense against pollution attacks in p2p systems with network coding. In: Proceedings of International Symposium on Network Coding (NetCod), pp. 1–7 (2011)
32. Li, Q., Chiu, D.M., Lui, J.C.: On the practical and security issues of batch content distribution via network coding. In: Proceedings of IEEE International Conference on Network Protocols (ICNP), pp. 158–167. IEEE, Santa Barbara (2006)
33. Li, Y., Yao, H., Chen, M., Jaggi, S., Rosen, A.: Ripple authentication for network coding. In: Proceedings of IEEE INFOCOM, pp. 1–9 (2010)
34. Lima, L., Médard, M., Barros, J.: Random linear network coding: a free cipher? In: Proceedings of International Symposium on Information Theory (ISIT), pp. 546–550 (2007)
35. Lima, L., Gheorghiu, S., Barros, J., Médard, M., Toledo, A.L.: Secure network coding for multi-resolution wireless video streaming. *IEEE J. Sel. Areas Commun.* **28**(3), 377–388 (2010)
36. Möller, U., Cottrell, L., Palfrader, P., Sassaman, L.: Mixmaster protocol—version 2. IETF Internet Draft (2003)
37. Ozarow, L., Wyner, A.: Wire-tap channel II. In: Proceedings of Advances in Cryptology, pp. 33–50 (1985)
38. Perrig, A., Canetti, R., Tygar, J.D., Song, D.: Efficient authentication and signing of multicast streams over lossy channels. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 56–73 (2000)
39. Reed, M.G., Syverson, P.F., Goldschlag, D.M.: Anonymous connections and onion routing. *IEEE J. Sel. Areas Commun.* **16**(4), 482–494 (1998)
40. Reiter, M.K., Rubin, A.D.: Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.* **1**(1), 66–92 (1998)

41. Rennhard, M., Plattner, B.: Introducing morphmix: peer-to-peer based anonymous internet usage with collusion detection. In: *Proceedings of ACM Workshop on Privacy in the Electronic Society*, pp. 91–102 (2002)
42. Rennhard, M., Plattner, B.: Practical anonymity for the masses with morphmix. In: *Financial Cryptography*, pp. 233–250. Springer, Heidelberg (2004)
43. Shields, C., Levine, B.N.: A protocol for anonymous communication over the internet. In: *Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS)*, pp. 33–42 (2000)
44. Silva, D., Kschischang, F.R.: Security for wiretap networks via rank-metric codes. In: *Proceedings of International Symposium on Information Theory (ISIT)*, pp. 176–180 (2008)
45. Vilela, J.P., Lima, L., Barros, J.: Lightweight security for network coding. In: *Proceedings of IEEE International Conference on Communications*, pp. 1750–1754 (2008)
46. Wang, J., Wang, J., Lu, K., Xiao, B., Gu, N.: Optimal linear network coding design for secure unicast with multiple streams. In: *Proceedings of IEEE INFOCOM (2010)*
47. Wicker, S.B., Bhargava, V.K.: *Reed-Solomon Codes and Their Applications*. Wiley-IEEE Press, New York (1999)
48. Wu, X., Li, N.: Achieving privacy in mesh networks. In: *Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 13–22 (2006)
49. Yeung, R.W., Cai, N.: Network error correction, I: basic concepts and upper bounds. *Commun. Inf. Syst.* **6**(1), 19–35 (2006)
50. Yu, Z., Wei, Y., Ramkumar, B., Guan, Y.: An efficient signature-based scheme for securing network coding against pollution attacks. In: *Proceedings of IEEE INFOCOM*, pp. 1409–1417 (2008)
51. Yu, Z., Wei, Y., Ramkumar, B., Guan, Y.: An efficient scheme for securing xor network coding against pollution attacks. In: *Proceedings of IEEE INFOCOM*, pp. 406–414 (2009)
52. Yun, A., Cheon, J.H., Kim, Y.: On homomorphic signatures for network coding. *IEEE Trans. Comput.* **59**(9), 1295–1296 (2010)
53. Zhang, Z.: Network error correction coding in packetized networks. In: *Proceedings of IEEE Information Theory Workshop*, pp. 433–437 (2006)
54. Zhao, F., Kalker, T., Médard, M., Han, K.J.: Signatures for content distribution with network coding. In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pp. 556–560 (2007)

Security in Network Coding

Zhang, P.; Lin, C.

2016, XI, 98 p. 34 illus., 18 illus. in color., Hardcover

ISBN: 978-3-319-31082-4