ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 9 September 2013 Page No. 2779-2782

Security Analysis Using Batch Verification Scheme

S.Alonshia, K.Ravikumar
Research scholar Department of Computer Science,Tamil University.
E-mail: elayabiotech@yahoo.com
Asst.Professor Department of Computer Science,Tamil University
E-mail:ravikasi2001@yahoo.com

Abstract

Network coding is seen as a promising technique to improve network throughput. In this paper, study two important problems in localized network coding in wireless networks, which only requires each node to know about and coordinate with one-hop neighbors. In particular, establish a condition that is both necessary and sufficient for useful coding to be possible. Content verification is an important and practical issue when network coding is employed. When random linear network coding is used, it is infeasible for the source of the content to sign all the data, and hence, the traditional "hash-and-sign" methods are no longer applicable. However, this technique is difficult to be applied to network coding designed for high calculation and message overhead. It explores this issue further by carefully analyzing different types of overhead, and proposes methods to help reducing both the computational and communications cost, and provide provable security at the same time. It show this condition is much weaker than expected, and therefore permits a change of coding schemes to suit different network conditions and application preferences. Based on the understanding establish, able to design a robust coding technique called loop coding that can improve network throughput and TCP throughput simultaneously.

Keywords: Content distribution, security, verification, network coding, loop coding

1. Introduction

To improve network throughput, the idea of network coding has been proposed for forwarding nodes to mix the bits in forwarded packets. In this work, focus on one specific type of network coding in wireless networks, where we XOR packets for unicast rows. This type of network coding has recently received a lot of practical interest for its ease of implementation and the importance of unicast communication. The basic idea of network coding using the Alice-and-Bob scenario, where Alice wants to send packet P1 to Bob and Bob likes to directthe packet P2 to Alice. They rely on a relay in the middle to exchange packets. In the terminology of network coding, a non-encoded original packet is referred to as a native packet. Network coding is aboutwhat packet should the relay transmit, in order for the nativepackets to be obtained by their intended receiver. An imperativeconcern in appliedfat content transfer in a fully distributed environment is presence of link failures, transmission errors , software and hardware faults, and even how tomaintain the integrity of the data ,
in themalicious attackers. If malicious attackers are able to modify the data in transmission, or inject arbitrary bogus data into the network, they may be able to greatly slow down the content distribution, or even prevent users from getting correct data entirely. In classical content distribution scenarios, data integrity can be checked using a "hash-and-sign" paradigm, where the source employs a collision resistant hash function h to compute hash values of the original data and signs the hash value using a digital signature scheme S with a signing key k. The signature is then used to verify received data Y. However, as can see later, such methods are not applicable in practical network coding-based content distribution schemes.

2. Network Coding

Network coding is a novel mechanism that promises optimal utilization of the resources of a network topology. With network coding , everytransmitted packet is a linear combination of all or a subset of the packets available at the sender (similar to X-OR multiple packets). Observe that encoded packet can be further combined to generate new linear combination. The original information can be reconstructed after receiving enough linearly independent packets. This is of great use in large scale distributed systems, such as P2P networks, where finding the proper scheduling of information across the overlay topology is very difficult. Compared to traditional approaches, network coding makes optimal use of the available network resources without the need for sophisticated scheduling algorithms and provides a high degree of robustness, even if nodes suddenly depart the system or if decisions are based only on partial information. A network coding based P2P file distribution

system in C#. The content distribution system consists of three types of participants: more than one peer networks, a administrator, and a logger. Peers are sources and sinks for content data. Peers exchange encoded information with each other in units that the call blocks. Gratified is sownhooked on the system by a special peer, which it call server. Peers that finish the download, but remain in the system are called seeds. The registrar enables peer discovery. The active peers periodically report to the registrar and the registrar provides a random subset of the active peers to knobs that have too few nationals. The logger is an aggregation point for peer and registrar trace messages. Each peer in this system generates detailed statistics to the logger; using those statistics it can able for performing in the depth evaluation of the distribution. The patrician is the highest complex of the three objects, and its functionality is divided into two components: network transport and content manager. The network transport maintains connections to other peers for transferring blocks. It uses two connections per pair of nodes (one for each direction). Each peer maintains 48 connections to other peers. Peers periodically drop a neighbor at random, encouraging cloud diversity and mitigating formation of isolated peer islands. The content manager encodes, decodes, validates, and persists content data. In our experiments, the file is divided into 10002000 original blocks; all transferred blocks can be expressed as combinations of the original blocks. To ensure low encoding and decoding times, it have grouped blocks into so called segments or generations, where only blocks of the same generation are combined. This approach, which it call Group Network Coding, results in more efficient decoding while retaining the network coding advantages. The encoding/decoding operations take place in a Galois Field.

3. Threat Model

In the network protocol stack, network coding schemes operate between the wireless link layer and higher layer protocols (such as TCP). Therefore, it is possible to design some robust coding schemes to mask the underlying link layer loss rate from higher layer protocols. Given the dual importance of reducing packet loss rate on improving both network throughput and TCP throughput, an ideal solution would be to design a practical network coding scheme that can effectively reduce packet loss rate, without incurring additional communication cost.

3.1 Entropy Attacks

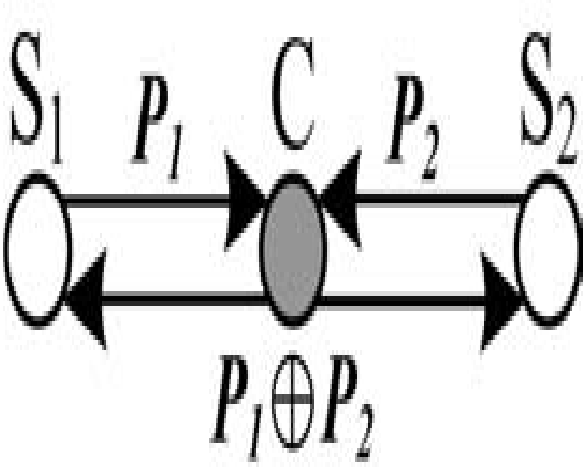
With network coding a node does not need to worry about how to pick the block to transmit to another node; it combines all its available blocks. Though, encoded blocks are only useful towards a given node if they carry new information, i.e. they are innovative. Determining innovation needs to take into account the coefficient vector used to generate the block; the encoded block downloaded can be different to each of the locally available encoded blocks, but, still, if its coefficient vector will be printed as a linear grouping of the routes of the nearbyoffered blocks, before it is not advanced. Using encoded blocks, would be very easy for an attacker to send non-innovative blocks that are trivial linear combinations of already existing blocks at the recipient. We call this an entropy attack since malicious users try to decrease the entropy or diversity in the system, reducing the opportunities that nodes have for making download progress and thus the rate of the system. Another side effect of such attacks is that a malicious user can easily become a free-rider even when incentive mechanisms like tit-for-tat are used by basically sending non-informative data and getting useful data in return. To solve this problem we ensure that each node, prior to downloading a block, first downloads the coefficient vectors of all the blocks in the neighborhood. By using the neighbors' coefficient vector and its own coefficient vectors, a given node then calculates the rank of the combined matrices and determines which neighbors can provide innovative blocks and moreover how many blocks they can provide. Rather than checking all the vectors from a potential sender, an substitute and frequently inexpensive method is having the sender create a random linear arrangement of all its coefficient vectors. The receiver forms whether the engendered coefficient vector can be articulated as a linear permutation of its coefficient vectors. If it cannot be written as a linear combination, then the sender has at least one innovative block that the receiver can download. If it can, then either the sender does not have any innovative blocks, or the random linear combination produced by the server fails to prove that the sender has indeed innovative information. This latter case is very rare and we can ignore such events. Observe that the transfer of the coefficient vectors generates little overhead meanwhile the size of every vector will adopt in a pair of packets, whereas the size each block is in the order of several hundreds of Kbytes.

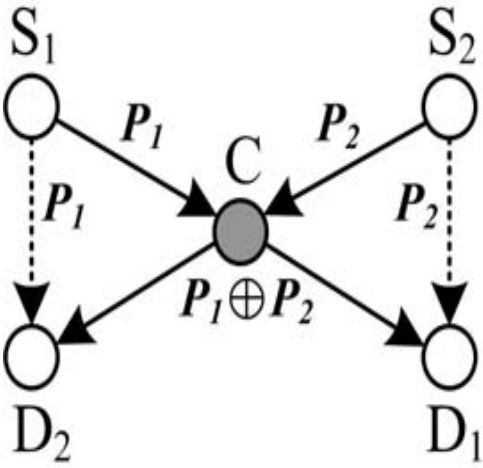
3.2 Jamming Attacks

In large scale P2P distributed systems, there exists the possibility that some nodes are malicious and inject bogus packets in the network to jam the download. Jamming attacks happen when a malicious node sends a couple of an encrypted block and a coefficient vector where any one doesn't carry the most valid information. The receiver will obtain corrupted information, and, if the receiver use this data to generate encrypted blocks, this can insert (involuntarily) the most degraded blocks in this network. Since receivers have limited bandwidth, they would clearly benefit from a mechanism that detects cheating as it happens, so they can terminate connections to bad neighbors and seek out honest nodes elsewhere in the network. Another alternative is to wait for the receiver to finish the download and try to decode the full file. However, if the file is infected with bad blocks, it is very difficult to identify such bad blocks at decoding time. Downloading extra blocks and performing multiple decoding operations with different combinations of blocks in an attempt to reconstruct a valid file has prohibitive cost. This is clearly unacceptable, especially for large downloads where the cost of multiple decoding becomes prohibitive. One bad block should not ruin hundreds or thousands of valid ones. To protect clients against jamming attacks, P2P cooperative systems require some form of source verification. That is, downloader's need a way to verify individual check blocks. Furthermore, this verification should work whether or not the original publisher is online. In standard P2P systems, this is obtained by hashing each block and allotting the block hashes that is taken from a central trusted publisher. Comparing the hash of every downloaded block for the conforming hash

function that is given by the publisher, a node can quickly check whether a block is valid or not. When blocks are encoded only at the server, or at a limited set of servers, then the standard way to prevent an attack by a malicious user injecting bad data into the system is to involve that legal senders sign all their encoded packets cryptographically. However, using network coding, jamming attacks are particularly serious because undetected malicious blocks will be used to generate more malicious blocks, and quickly every block transmitted in the network is corrupted. Observe that each encoded block is unique and cannot be signed by a trusted authority, like for example the server. Thus, to prevent a jamming attack in an open system that uses network coding, one would need a hashing scheme such that the hash of an encrypted packets can be easily derived from the hashes of the unique packets and that is obtained from the coefficient vector that designates the encoding. Homomorphism hashes have this property and are described in the following section.

Diagram





4. Batch Verification

The Baseline Batch Verification Scheme
To reduce the computational cost of the verification, a batch of packets can be verified at the same time. In particular, after a node has received b packets ðð y1; c1Þ; ðy2; c2Þ;. . . ; ðyb; cbÞÞ (not necessarily from the same source), the node containerauthenticate all the packets as follows:
1. Randomly choose b numbers r1; . . . ; rb 2 ZZq.
2. Compute w ¼ Pb i¼1 riyi mod q.
3. Compute v ¼ Pb i¼1 rici mod q.
4. Verify the integrity of the packet ðw; vÞ using the basic integrity verification scheme. Due to the homomorphism property of the hash function, this will see that the batch verification in the above Step 4 fails, then at least one of the packets is corrupted. However, the set of the packets permit the verification, it is still possible that some packets are corrupted but may well not be spotted by the verification algorithm. Therefore, it needs to investigate the security more carefully.

5. The Secure Random Checksum Scheme

An alternative to the expensive cryptographic hash function is called SRC. Their focalindication is as following. Before the authentictransferring commences, every node rescues a checksum for bothslab of data from the source via a secret channel. Each one checksum is calculated as a random combination of entirely the sub blocks in a block, and the quantitiesis foundnot the same for each node. These checksums are also homomorphism, so that they can be used to checked the integrity of any established packet in a way similar to homomorphism hash functions. Advancedone linear arrangements are intricate in the computation of these checksums, the authenticationcontainer be sameable. It denotes that in this scheme, every node wants to download a group of separate checksums straight from the source. It provides a integrated downloading situation with a smaller content size (which is the size of the checksums), which may lead to two problems. Firstly, this carriages limits on the reliability of the scheme, since the source could be overcome by the requests to download checksums. Whereas in the case of homomorphism hashes, although the hash values have larger sizes, it is not necessary to download them straight taken from the source but they will be instead taken from peers, and no additional secure channel is needed. Secondly, the source is essential to be online till all the nodes have received checksums from it, which makes it difficult for dynamic networks where nodes are frequently leaving and joining the network. To some extent, the use of such checksums weakens the potential advantages one could expect from a distributed content distribution scheme.

6. Conclusions

To improve system throughput or P2P networks to improve overall system efficiency. In this paper, investigate the security and efficiency issues in large content distribution based on network coding. Consider the problem of on-the-fly verification of the integrity of the data in transit. Although a previous scheme based on homomorphism hash functions is applicable, it was mainly designed for server side coding only, and will be much less efficient when it is applied on random network coding. Propose a new on-the-

fly verification scheme based on a faster homomorphism hash function, and proved its security. Consider the computation and communication cost incurred during the content distribution process. It can identify various sources of the cost, and investigate ways to eliminate or reduce the cost. In particular, propose a sparse variant of the classical random linear network coding, where
only a small constant number of blocks are combined each time.

7. References

[1] B. Fan, J.C.S. Lui, and D.-M. Chiu, "The Design Trade-Offs of BitTorrent-Like File Sharing Protocols," IEEE/ACM Trans. Networking, vol. 17, no. 2, pp. 365-376, Apr. 2009

[2] J. Le, J.C.S. Lui, and D.-M. Chiu, "DCAR: Distributed Coding- Aware Routing in Wireless Networks," IEEE Trans. Mobile Computing, vol. 9, no. 4, pp. 596-608, Apr. 2010.

[3] J. Le, J.C. Lui, and D.-M. Chiu, "On the Performance Bounds of Practical Wireless Network Coding," IEEE Trans. Mobile Computing, vol. 9, no. 8 pp. 1134-1146, Aug. 2010.

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the air: Practical wireless network coding. In ACMSIGCOMM, 2006

[5] C. Gkantsidis and P. Rodriguez, "Cooperative Security for Network Coding File Distribution," Proc. IEEE INFOCOM, pp. 1-13, Apr. 2006.