Providing Security and Efficiency for Content Distribution Using Network Coding In Wireless Sensor Networks

Shyleshwari. M. Shetty
P.G.Schloar
Computer science and Engineering
APS College of Engineering
Bangalore, India
shyleshwari.m.shetty@gmail.com

Abhijit Das
Senior Lecturer, Dept. of ISE
APS College of Engineering
Bangalore, India
abhijit.tec@gmail.com

***Abstract*—Content distribution using network coding has received a lot of attention. When we directly apply network coding technique it may be insecure because attackers can inject bogus data to corrupt the content distribution process so that information dispersal will become slow or network resource may get deplete. Hence, content verification is an important and practical issue when we make use of network coding. When random linear network coding is used, it is infeasible for the source of the content to sign all the data .Hence, the traditional hash and sign methods are no longer applicable. A new on-the-fly verification technique has been proposed which uses a classical homomorphic hash function. However, this technique is difficult to be applied to network coding. Because it has high computational and communication overhead. This issue is further explored and proposes a methods that helps to reduce both the computational and communication cost and provide the security at the same time.**

Keywords— Content Distribution; Content Verification; Network Coding; Bogus Data.

I. INTRODUCTION

For the past few years there has been an increasing interest on the application of network coding on file distribution. Various researchers have considered the benefit of using network coding on P2P networks for file distribution and multimedia streaming and also on millions of PCs around the Internet for massive distribution of new OS updates and software patches. The main focus here is the security of content distribution schemes using network coding and how to achieve the security at the same time.

Network Security is used to provide security to the authorized data which is being distributed from source to destination in the network. It also prevents unauthorized access of data by developing a secure network using security service like access, confidentiality, authentication, integrity, non-repudiation. Some of common internet attack methods used to modify the authorized data are eavesdropping, viruses, worms, Trojan, IP spoofing, denial of service.

To prevent data from such attacks technologies like cryptographic systems, firewall, intrusion detection systems, anti malware software and Scanners and secure socket layer is used. Current development in the network security is the biometrics and smartcard which greatly reduces the unauthorized access of secure systems.

An important issue in practical large content delivery in a fully distributed environment is how to maintain the integrity of the data in the presence of link failures, transmission errors, software and hardware faults, and even malicious 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.

A. characteristics and applications of wsn's

The characteristics of WSN are- Power consumption constrains for nodes using batteries or energy harvesting, Communication failures, Ability to cope with node failures, Mobility of nodes, Dynamic network topology, Heterogeneity of nodes, Scalability to large scale of deployment, Ability to withstand harsh environmental conditions, Easy of use Unattended operation.

Applications of WSN are:

- Industrial automation
- Automated and smart homes
- Video surveillance
- Traffic monitoring
- Medical device monitoring
- Monitoring of weather conditions
- Air traffic control
- Robot control.

II. DEFINITIONS

Content Distribution is the process of transmitting the messages or data from source to destination. Content Distribution is the act of sharing or circulating content with other websites, directories, or users. Content Distribution is a great means for product companies to circulate their products through various online means.

Content verification means verifying the contents with its strength to check whether the received content is modified by unauthorized users.

Network coding is the set of techniques or algorithm for giving security during transmission via networks. Network coding is a technique which can be used to improve a network's throughput, efficiency and scalability, as well as resilience to attacks and eavesdropping, as compared to traditional methods of OSI model or TCP/IP model.

Bogus data is to insert fake data to the original data by unauthorized users.

III. Related Work

The maximum capacity between a source and a sink connected through a network is the same as the maximum network flow f between them. When the network can be viewed as a directed acyclic graph with unit capacity edges, f is also the min-cut of the graph between the source and the sink. However, when there is a single source and multiple sinks, the maximum network flow f may not be achieved. If the nodes in a network can perform coding on the information they receive it is possible for multiple sinks to achieve their maximum network flow bound simultaneously through the same network. Then it becomes possible to achieve the theoretical capacity bound if one allows the network nodes on the path to perform coding, instead of just the conventional tasks of routing and forwarding.

Later, Li et al. showed that, although the coding performed by the intermediate nodes does not need to be linear, linear network codes are indeed sufficient to achieve the maximum theoretical capacity in acyclic synchronous networks. In their settings, each node computes some linear combination of the information it receives from its upstream nodes, and passes the results to its downstream nodes. However, to compute the network code (i.e., the correct linear combinations) that is to be performed by the nodes, the topology of the network has to be known beforehand, and has to be fixed during the process of content distribution. Furthermore, their algorithm is exponential in the number of edges in the network.

Koetter and Medard considered the problem of linear network coding. They improved and extended the results by Li et al. and considered the problem of link failures. They found that a static linear code is sufficient to handle link failures, if the failure pattern is known beforehand. However, as mentioned by Jaggi et al. the code construction algorithm proposed by Koetter et al. still requires checking a polynomial identity with exponentially many coefficients.

Jaggi et al. proposed the first centralized code construction algorithm that runs in polynomial time in the number of edges, the number of sinks, and the minimum size of the min-cut. The network coding does not improve the achievable transmission rate when all nodes except the source are sinks, finding the optimal multicast rate without coding is NP-hard.

With the popularity of P2P networks researchers are beginning to consider the problem of on-the-fly Byzantine fault detection in content distribution using random network coding. The verification techniques proposed by Krohn et al. can be employed to protect the integrity of the data without the knowledge of the entire content. The verification techniques were originally developed for content distribution using rateless erasure codes and were based on homomorphic cryptographic hash functions.

Random network coding was proposed by Ho et al as a way to ensure the reliability of the network in a distributed setting where the nodes do not know the network topology, which could change over time. In their setting, each node would perform a random linear network coding, and the probability of successful recovery at the sinks can be tightly bounded. Chou et al, proposed a scheme for content distribution based on random network coding in a practical setting, and showed that it can achieve nearly optimal rate using simulations. Recently, Gkantsidis and Rodriguez proposed another scheme for large scale content distribution based on random network coding. They show by simulation that when applied to P2P overlay networks, using network coding can be 20 to 30 percent better than server side coding and 2 to 3 times better than uncoded forwarding, in terms of download time.

IV. Problem Statement And Proposed System

In existing system, content is sent from the source to destination. By applying hash technique we get hashed content and key. Same key is used for the same content every time. Hashed content is sent to the destination through the centralized server. If the destination does not have the capacity of storing the content which is received from the source then both transmission rate and delay will be too high. Thus by sending same key for the same content unauthorized users easily modify the content.

In proposed system, three techniques to maintain integrity of the content being distributed from the source to the destination is proposed. First Random linear network coding is used to split the content and to store the content in different storage locations randomly. Thus by splitting the content the transmission rate and delay will be less and network traffic will also be avoided. Homomorphic hash function is the second technique to hash the splitted content and to generate the keys randomly. Hashed content is randomly distributed to the distributed network. Hashed content strength and keys is stored in the source.

Finally on-the-fly verification technique is used in which three methods are used. Data verification verifies hashed content and hashed content strength. Block by block downloading downloads hashed content from distributed network, random keys and hashed content strength from the source. Using keys we dehash the hashed content to get original splitted content Reconstruction reconstructs the splitted content which is received randomly to get the original content sent from the source.

V. A Homomorphic Hash Function Based on VSH

The proposed basic scheme is based on the nontrapdoor variant (VSH-DL) of the Very Smooth Hash (VSH) function. The rationale behind VSH and its variants is that using smaller primes as group generators would greatly improve the computational efficiency of hash functions that involve many exponentiations. The VSH functions, however, do not have the homorphic property that would be required by the on-the-fly verification process. The homomorphic property is obtained by rearranging the order of the bits in the input message blocks before applying VSH-DL.

A Homomorphic VSH-DL

Let p be a large strong prime such that there is another large prime q divides p-1. That is, there is a positive integer α such that p=αq+1. As a result, there exist a multiplicative subgroup G in Z_p* with order q. Furthermore for any x ∈ $Z_{p,}$, let $y=x^{\alpha}$ mod p, if y ≠ 1 mod p, y must be in G.

Let $p_1, \ldots\ldots, p_m$ be m prime numbers, such that $m < \gamma^c$ for some constant c. In other words, m is bounded by some polynomial in γ. In practice, we can choose those primes to be the m smallest prime numbers, such that $p_i^{\alpha} \neq 1$ mod p. That is, we can choose $p_1=2$, $p_2=3$ and skip those primes whose order is not q. When q is much larger than α(e.g α=2) the probability that a random small prime p_i satisfies the condition $p_i^{\alpha} \neq 1$ mod p is high. Assume that a message is a vector of the form: x =(x_1,…..x_m) m where $x_i \in Z_q$ for $1 \leq i \leq m$.

The hash of x is computed as

$$H(x)= \prod_{i=1}^{m} p_i^{\alpha x_i} \bmod p$$

VI. The Basic Integrity Verification Scheme

A. The Basic Scheme

The proposed scheme consists of two algorithms, namely, the encoding algorithm where the original data are prepared for distribution and the verification algorithm, which is used by individual nodes to verify the integrity of the received data.

B. Encoding

Let the parameters p, q, m and p_1 ,…, p_m be chosen as in Section 7.1Given any binary string X, let n be the smallest positive integer such that $|x| < mn(\gamma - 1) - 1$. Assume that $n < poly(\gamma)$ for some positive polynomial poly. We also assume that X is compressed, such that the bits are random.

In this way, we can always pad the original X properly (e.g., with a one followed by zeros) such that the result can be divided into small pieces of γ - 1 bits each. In other words, we can always encode the data into the form x=(x_1,…..,x_n) where $x_i=(x_{i,1},\ldots,x_{i,m})^T$ and each $x_{i,j}$ is of length γ - 1, and can be considered as an element in Z_q for all $1 \leq i \leq n$ and $1 \leq j \leq m$.

We will call x_i as the ith block, and each $x_{i,j}$ as the jth subblock of the ith block. Now, given X, the encoder computes

$$h_i = H(x_i) = \prod_{j=1}^{m} p_j^{\alpha x_{i,j}} \bmod p$$

for each $1 \leq i \leq n$.

C. Basic Verification Algorithm

During verification, each network node is given a packet (x,c) and system parameters. In the case where this packet is not tampered with, c = ($c_{1,\ldots,}c_n$) are the coefficients where each $c_i \in z_q$, x is the linear combination $x = \sum_{i=1}^{n} c_i x_i \bmod z_q$.

Each node can verify the integrity of the packet as following:

- Compute the hash value $H_1 = H(x)$
- Compute $H_2 = \prod_{i=1}^{n} h_i^{c_i} \bmod p$.
- Verify that $H_1 = H_2$.

It is worth to note that once an intermediate node has received the parameters p, q, and m, it will be able to compute α and $p_1,\ldots\ldots,p_m$ locally. Therefore, those prime bases do not need to be distributed. The hash values, however, still need to be distributed. Nevertheless, we will see that in any reasonable setting the communication overhead is low.

VII. Batch Verification

A. 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 $((y_1,c_1),(y_2,c_2),\ldots.(y_b,c_b))$ the node can verify all the packets as follows:

- Randomly choose b numbers $r_1,\ldots.,r_b \in z_q$
- Compute $w = \sum_{i=1}^{b} r_i y_i \bmod q$.
- Compute $v = \sum_{i=1}^{b} r_i c_i \bmod q$
- Verify the integrity of the packet (w,v) using the basic integrity verification scheme.

Due to the homomorphic property of the hash function, we can see that if the verification in Step 4 fails, then at least one of the packets is corrupted. However, if the batch of packets pass the verification, it is still possible that some packets are corrupted but could not be detected by the verification algorithm. Hence, we need to analyze the security more carefully.

VIII. Sparse Random Linear Network Coding

The computation overhead involved in the content distribution consists of two parts. The first part is the cost due to the verification of the packets and the second part is the cost due to the need to compute random combinations of the data blocks.

The cost can be reduced through the use of more efficient hash functions and batch verification techniques. the second part of the cost also plays a very important role in practice, especially when the content is large (e.g., in the order of gigabytes), and it has a significant impact on the choice of parameters.

The method involves divide the content to be distributed into smaller trunks and random linear network coding is applied to each trunk of content independently. Although this method works in certain application scenarios, it does not address the problem directly but instead avoids high computation overhead by applying random linear network coding to smaller problem instances.

Hence, this strategy may lose certain benefits from network coding. For example, when a node sends data to its downstream nodes, it has to decide which trunk to send. If the algorithm to make such decisions is not designed properly, it may result in a situation where a certain trunk cannot be reconstructed after a few key nodes have left the network.

A simple yet powerful alternative to avoid high computation cost when computing the random combinations is proposed. This method is referred as Sparse Random Linear Network Coding.

The idea is that, instead of computing a random combination of all the n data blocks, we can instead randomly select only θ of them and compute a random combination of only those θ blocks. When a node A needs to send a packet (x,c) to its downstream node, it performs the following steps:

- Randomly choose θ packets from the random combinations received by A so far. Let these packets be $(x_1,c_1),\ldots,(x_\theta,c_\theta)$.
- Randomly choose $r_1,\ldots, r_\theta \in z_q$
- Compute packet (x,c) as

$$x = \sum_{i=1}^{\theta} r_i x_i \quad , \quad c = \sum_{i=1}^{\theta} r_i c_i$$

Also, we require the source node to be more powerful than other nodes and still send random combinations of all n blocks. It is clear that each packet being sent over the network would still be quite random, and allow high probability of reconstruction at the receivers.

The probability of successful delivery is very high even with small constant θ. Therefore, the computation overhead due to the computation of random combinations can be made independent of the number of blocks, which greatly relaxes the constraints that need to be considered for practical systems.

IX. Computation Overhead

The computation cost involves both the computation of random combinations and the hash values. When sparse random linear coding is applied, for each combined block we only need to compute the combination of θ blocks, which makes the computation of combinations much more efficient than that of the hash values.

In Fig 1, the throughput of the computation of H is computed as the size of a data block divided by the average time it takes to compute a hash value for the block. The values are taken as the average 25 random instances. we can see that throughput increases when the number of subblocks per block is increased.

Even with relatively small values of λ and γ(say λ=512 and γ=400), the computation of H_1 cannot be very efficient compared with common hash functions such as SHA-1. With carefully designed precomputation methods the throughput can be increased by a small constant factor r at the price of a storage requirement that is 2^r times that without precomputation.

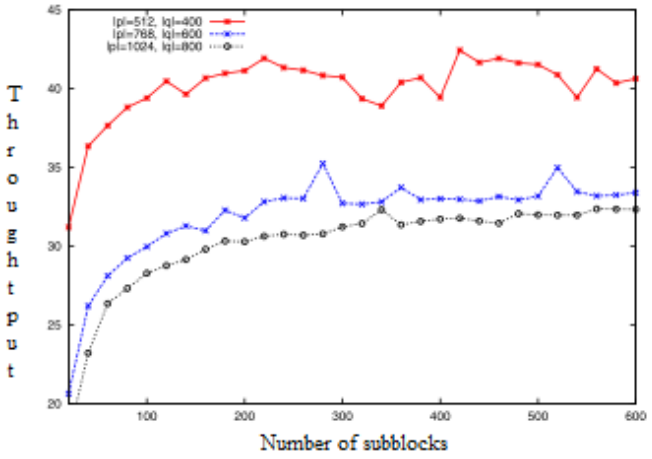

Fig. 1. Computation Efficiency of H_1

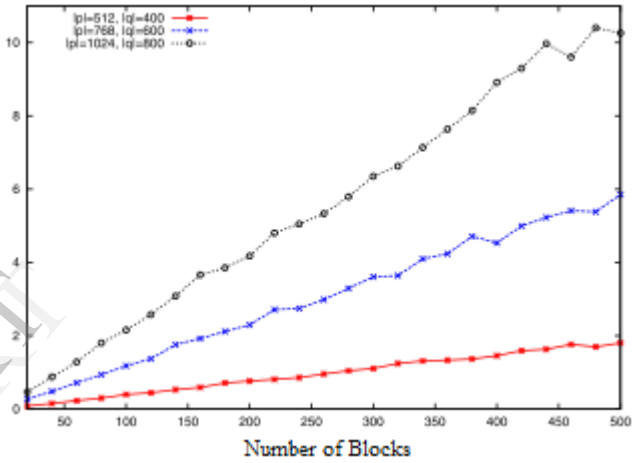

Fig. 2. Computation Efficiency of H_2

In Fig 2, the throughput under different security parameters is the gradient of different curves in the figure. For example, for p=512 and q=400, when n= 500, the time it takes to compute H is about two seconds. This translates to a throughput of about 12.5 kilobytes per second, which is much lower than 40 kilobytes per second. For p=1024 and q=800 similar calculations show that the throughput is greatly reduced to about 5 kilobytes per second.the advantage is that the efficiency is not reduced as much when the parameters are increased for stronger security. The computational advantage is mainly due to the use of deterministically chosen small primes as the bases for exponentiations, which is the rationale behind the design of the VSH scheme.

X. Conclusion

The problem of on-the-fly verification of the integrity of the data in transit is considered. Although a previous scheme based on homomorphic 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. A new on-the-fly verification scheme based on a faster homomorphic hash function is proposed and proved its security.

The computation and communication cost incurred during the content distribution process is considered and identify

various sources of the cost and investigate ways to eliminate or reduce the cost. A sparse variant of the classical random linear network coding is proposed where only a small constant number of blocks are combined each time.

Experiments are conducted to examine the efficiency of the proposed hash function, as well as the effectiveness of the proposed sparse random linear network coding. The results show that the new hash function is able to achieve reasonable speed, and the sparse variant performs just as well as the random network coding using typical parameters.

REFERENCES

[1] S. Acedanski, S. Deb, M. Medard, and R. Koetter, "How Good Is Random Linear Coding Based Distributed Networked Storage," Proc. Workshop Network Coding, Theory and Applications, Apr. 2005.

[2] P.A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," Proc. Allerton Conf. Comm., Control, and Computing, Oct. 2003.

[3] C. Gkantsidis and P.R. Rodriguez, "Network Coding for Large Scale Content Distribution," Proc. IEEE INFOCOM, pp. 2235-2245, 2005.

[4] M. Wang, Z. Li, and B. Li, "A High-Throughput Overlay Multicast Infrastructure with Network Coding," Proc. Int'l Workshop Quality of Service (IWQoS), 2005.

[5] Y. Zhu, B. Li, and J. Guo, "Multicast with Network Coding in Application-Layer Overlay Networks," IEEE J. Selected Areas in Comm., vol. 22, no. 1, pp. 107-120, Jan. 2004.

[6] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network Information Flow," IEEE Trans. Information Theory, vol. 46, no. 4, pp. 1204-1216, July 2000.

[7] S.R. Li, R.W. Yeung, and N. Cai, "Linear Network Coding," IEEE Trans. Information Theory, vol. 49, no. 2, pp. 371-381, Feb. 2003.

[8] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," IEEE/ACM Trans. Networking, vol. 11, no. 5, pp. 782-795, Oct. 2003.

[9] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain, and L.M. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," IEEE Trans. Information Theory, vol. 51, no. 6, pp. 1973-1982, June 2005.

[10] C. Gkantsidis and P. Rodriguez, "Cooperative Security for Network Coding File Distribution," technical report, Microsoft Research, 2004.

[11] T. Ho, B. Leong, R. Koetter, M. Me'dard, M. Effros, and D.R. Karger, "Byzantine Modification Detection in Multicast Networks Using Randomized Network Coding," Proc. IEEE Int'l Symp. Information Theory, 2004.

[12] C. Gkantsidis, J. Miller, and P. Rodriguez, "Anatomy of a P2P Content Distribution System with Network Coding," Proc. Int'l Workshop Peer-to-Peer Systems, Feb. 2006.

[13] T. Ho, R. Koetter, M. Me´dard, D.R. Karger, and M. Effros, "The Benefits of Coding over Routing in a Randomized Setting," Proc. IEEE Int'l Symp. Information Theory, 2003.

[14] M.N. Krohn, M.J. Freedman, and D. Mazie`res, "On-the-Fly Verification of Rateless Erasure Codes for Efficient Content Distribution," Proc. IEEE Symp. Security and Privacy, pp. 226-240, May 2004.

[15] M. Bellare, O. Goldreich, and S. Goldwasser, "Incremental Cryptography: The Case of Hashing and Signing," Proc. CRYPTO,1994.