International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

# Improved Performance Approach for P2P Networks with Network Coding

**Aparna Jumde[1], Shyamsundar Gupta[2]**

[1, 2] Pune University, Siddhant College of Engineering, Sudumbare, Pune, Maharashtra, India

**Abstract:** ***Network coding based applications are reprehensibly susceptible to malicious pollution attacks. Packets authentication schemes have been well-recognized as the most effective approach to address this security threat. However, existing packets authentication scheme i.e. Times keys scheme utilizes extra bandwidth for transmission of times keys. Bu tin this scheme replay attack can be launched by capturing the Time keys by the attacker, and attacker can introduce a malicious packet in network. As to save bandwidth we are using network coding this attack may get prorogated throughout the network and result into pollution attack. The solution to avoid the problems in Time keys scheme is based on the concept that the source and the destination must agree on the keys along with the network coding process. This scheme of agreeing on the keys is known as adaptive TESLA scheme. Using this proposed scheme the effect of pollution attack and collusion attack is measured in terms of real packets corrupted and dropped. The proposed TESLA scheme for network coding is also compared with times keys in terms of bandwidth usages.***

**Keywords:** Network Coding, Pollution attack, Collusion attack,TESLA, P2P Network

## 1. Introduction

Peer-to-peer (P2P) converged ubiquitous network has attracted more and more attention in the field of ubiquitous communication. Although P2P converged ubiquitous network has distinct advantages, it still faces great challenges in practical application. P2P network exploits a better share of available resources, such as storage space and upload bandwidth. It is important to efficiently use the buffer space allocated in each peer and improve the service availability of P2P network. However, peers may join or leave P2P networks optionally, which will introduce block reconciliation problem. Without adequate blocks diversity, some blocks may become so rare that a part of peers may have difficulty to locate those blocks or even suffer from starvation. Fortunately, P2P converged ubiquitous can be greatly benefited from network coding.

In P2P networks with random network coding, each peer breaks the conventional data transmission mechanism by allowing mixture of its data. Compared with the conventional way, network coding can reduce bandwidth consumption and raise service availability. Moreover, a recent research shows that network coding can lead to a significant blocks diversity and thus solve the block reconciliation problem. Not only reducing redundant data transfer, but also improving the robustness of P2P network has been shown as the benefits brought by network coding. However, P2P networks with random network coding are known to be susceptible to pollution attack. During such attacks, malicious peers inject polluted packets to their neighbors. Such pollution can rapidly propagate in the network, leading to substantially degraded performance due to the wasted bandwidth of corrupted blocks distribution.

Furthermore, since network coding allows peers to forward packets coded from their received packets, as long as a single input packet is corrupted, all output packets forwarded by a peer will be corrupted. As a result, a single corrupted block will pollute the whole network system and prevent peers from decoding the original blocks, thus degrading the performance of the whole system. Unless this problem is solved, network coding schemes may perform much worse than pure forwarding data transmission schemes in the presence of pollution attacks. Therefore, it is crucial to check coded packets whether they are corrupted before using them for encoding in network coding systems, and filter out polluted packets as early as possible.

### 1.1 Pollution attacks

Malicious nodes launch pollution attacks either by injecting spurious packets or by modifying their output packets to contain corrupted data. We say a packet (c, e) is a polluted packet if the vector e is not equal to the product of the original generation of packets {p1, p2, · · · , pn} and the global encoding vector c. i.e., the following in equation:

$$e \neq \sum_{i=1}^{n} cipi$$

### 1.2 Collusion attacks

Collusion attack means that malicious nodes collude to cheat more payment from source node. The main goal of collusion attacks is to destroy the cryptographic function of security scheme in network coding system. They attempt to compromise as many intermediate nodes as possible to obtain the confidential encryption information. If a node is compromised, attackers can read its memory and monitor all incoming and outgoing communications. Attackers can also collaborate to launch collusion attacks, making the security scheme easy to be destroyed.

## 2. Literature survey

So far, a number of schemes have been proposed to defend against pollution attacks in P2P converged ubiquitous network with network coding. These schemes can be categorized into two categories.

- **End-to-end schemes:** Polluted message is detected only at destination peers.
- **In-network schemes:** Detection is at all participating peers.

An end-to-end approach makes minimal changes to existing network coding algorithms, and only the source and destination peers are involved in performing computations to enable detection and correction of errors introduced by pollution attack. However, when adversaries locate their attacks at the bottleneck of the network, end-to-end schemes can lead to a worst-case view of adversarial attack. In network solutions use cryptographic approaches by which participating peers detect and drop polluted packets. However, in-network schemes are various from each other in efficiency and security, and there is no practical in network schemes so far. Previous solutions to the problem will be worse when facing a large number of colluding peers.
Ming He, Zhenghu Gong, Lin Chen, HongWang [1] in their work "Securing network coding against pollution attacks in P2P converged ubiquitous networks" proposed time keys, an in-network solution for network coding against pollution attacks along with arbitrary collusion among malicious peers. In our current paper work we explore this work & propose mechanisms to improve their approach.

Time keys scheme is a in-network security scheme basing on time and space properties of network coding, and time keys scheme is secure in collusion attacks. They verify the integrity of a block by checking if it belongs to the current subspace at the source. Although an attacker is able to generate corrupted packets that match its known time keys, it cannot convince other nodes to accept hem, as these packets will not be verified with the time keys known to the attacker, but with another new version time keys generated by the source at a time after the packets are received. For the timeliness of time keys, it is hard for malicious node to know the content of its neighbor, and in that case, it is hard to find a corrupted block that can pass the verification process. Their scheme does not assume the existence of a secret channel and it is based on pseudorandom generators, and our work is an innovative time and space based solution to frustrate pollution attacks with arbitrary collusion among malicious nodes.

Adrian Perrig Ran Canetti J. D. Tygar [2] Dawn Song in their work "The TESLA Broadcast Authentication Protocol" puts some points regarding TESLA. The security of TESLA relies on the following assumptions:

- The receiver's clock is time synchronized up to a maximum error of Δ, because of clock drift; the receiver periodically resynchronizes its clock with the sender.
- The functions F; F0 are secure PRFs, and the function F furthermore provides weak collision resistance**.**

TESLA is not a signature mechanism and does not provide non-repudiation, as anybody could forge "authentic" TESLA packets after the key is disclosed. However, in conjunction with a trusted time stamping mechanism, TESLA could achieve properties similar to a digital signature. Consider this setup: all nodes in the network are loosely time synchronized and all nodes in the network trust the time stamping server. The time stamping server timestamps all TESLA packets it receives. The time stamping server can broadcast the hooks to the trust chain authenticated with its TESLA instance. A judge who wants to verify that a sender sent packet P performs the following operations:

1) Receive the current value of the time stamping server's trust chain, ensure that it is safe, and wait for the TESLA key to authenticate it.
2) Based on the trust chain value, verify that packet P is part of the trust chain.
3) Verify that packet P was safe when the time stamping server received it (not necessary if the time stamping server only timestamps safe packets).
4) Retrieve key from the sender and verify it using the key chain commitment and disclosure schedule recorded by the time stamping server.
5) Verify that the authenticity of the packet, which implies that the correct sender must have generated the packet. TESLA and a time stamping server can thus achieve non-repudiation. This example also shows that the TESLA authentication can also be performed after the key is already disclosed, as long as the verifier can check that the packet arrived safely.

### 2.1 Motivation

Time based scheme has following problems.

1) Time keys can be captured & replay attack can be launched by the attacker.
2) Extra bandwidth is consumed for transmission of time keys & for the large network this will become a critical bottleneck.
3) This motivates us to design a new solution to avoid the problems in time key scheme.

## 3. Problem statement and solution strategy

This chapter discusses the problem statement, solution strategy, assumptions and objectives of the project.

### 3.1 Problem Statement

P2P networks with random network coding are known to be susceptible to pollution attack. During such attacks, malicious peers inject polluted packets to their neighbors. Such pollution can rapidly propagate in the network, leading to substantially degraded performance due to the wasted bandwidth of corrupted blocks distribution.

Furthermore, since network coding allows peers to forward packets coded from their received packets, as long as a single input packet is corrupted, all output packets forwarded by a peer will be corrupted. As a result, a single corrupted block will pollute the whole network system and prevent peers from decoding the original blocks, thus degrading the performance of the whole system Therefore, it is crucial to check coded packets whether they are corrupted before using them for encoding in network coding systems, and filter out polluted packets as early as possible. Time based scheme was

proposed to solve this problem. But the time based keys can be captured & replayed in the network.

### 3.2 Solution Strategy

The Solution is based on the same concept of time keys but without transmitting the time keys the source & the destination nodes must agree on the keys. This scheme is referred as Adaptive TESLA. Each node with a seed key generates N number of keys using one way hash function. Let seed key be S1. Subsequent keys generated by one way hash chain is
S2, S3, S4, S5 ….. SN

All the nodes are assumed to be time synchronized. Every time once nodes use the key from SN, SN-1,… S2 (i.e. in the reverse order).

The network coding vectors are encrypted & also digital signature is prepared using the Key S for that time interval. After the time interval using S1 & current time stamp, second generation keys are prepared. When any corrupt packets are launched in the network, at the receiver, the digital signature is verified using the Key S for that time interval & dropped if the signature match failed. This way we can avoid the pollution & colluding attack without consuming extra bandwidth. Also replay attack cannot be launched in our approach because a key at time instant is not same as any, since keys are regenerated at each generation.

### 3.3 Objective

The main objectives of the project are as follows:
1) Measure the effect of pollution attack & colluding attack on the proposed solution in terms of number of real packets corrupted & dropped.
2) We will also measure the bandwidth consumed in our approach.
3) The proposed Adaptive TESLA scheme is compared with time key solution in terms of bandwidth usage.
4) Measure the effect of replay attack in the Adaptive TESLA & time key solution in terms of number of real packets corrupted & dropped.

## 4. Mathematical model

To begin, consider an acyclic network $(V,,c)$ with unit capacity edges, i.e., $c\ e$ =1 for all $e\in E$, meaning that each edge can carry one symbol per unit of time. Assume also that each symbol is an element of a finite field $F$. Let there be a single sender $s\in V$ and a receivers $T\subseteq V$. Let Let $x1,...,$xn be the h symbols that we wish to unicast from $s$ to $T$ in each unit of time. For each edge $e$ emanating from a node $v$, let $ye$ denote the symbol carried on $e$. The symbol $y$ , regarded as an element of the finite field $F,,$ can be computed as a linear combination of the symbols $y(e')$ on edges $e''$ entering node $v,,$ namely,
$y\ e = \beta e'e'\ e\ y(e')$.

The coefficients of the linear combination form a vector $\boldsymbol{g}\ e = \beta e'$ , known as the local encoding vector on edge $e$. The length of this vector is the number of edges $e$ 'entering $v$. The local encoding vectors on edges $e$ leaving $v$ characterize the network functions performed at $e$.
The coefficients of this linear combination form a vector $\boldsymbol{g}\ e =[g1\ e\ ,..,g$h $e\ ],,$ known as the global encoding vector on edge $e$. The global encoding vector $(e)$ represents the code symbol $y(e)$ in terms of the source symbols $x1,...,$xn. It is easy to see that the global encoding vectors themselves can be computed recursively as $\boldsymbol{g}\ e = \beta e'e'\ e\ (e')$ using the coefficients of the local encoding vectors $\boldsymbol{\beta}(e)$

Suppose now that a receiver $t\in T$ receives code symbols $y\ e1$ $,...,y(e$h$)$on edges $e1,...,e$h entering $t$. The received code symbols can be expressed in terms of the source symbols as

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}$$

where the i[th] row of the matrix $Gt$ is the global encoding vector associated with edge $ei$ entering receiver $t$. Receiver $t$ can therefore recover the h source symbols by inverting the matrix $Gt$ and applying the inverse to its received code symbols. The sender and the receiver both are time synchronized, the sender splits the time into even intervals Ii. The duration of each time interval is denoted by Tint and the starting time of the interval Ii is Ti. Hence
Ti =T0 + i * Tint

Before sending the first generation of messages, the sender determines the sending duration, the interval duration, and N i. e. number of keys of the key chain. The sender picks the last key KN of the key chain and pre-computes the entire key chain using a pseudo-random function F. Each element of the chain is defined as Ki= F (Ki +1). Along with this keys the function of coding also performed using function

$$z = \sum_{j=1}^{m} cjzj$$

Where cj is the global encoding vector and zj is packet from particular generation of size m. The sender sends encoded vector along with the one of the key from the key chin generated by the TESLA key manager.

## 5. System Architecture

System architecture is the conceptual design that defines the structure and behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system. The overall architecture of the proposed system is as shown in figure 1.

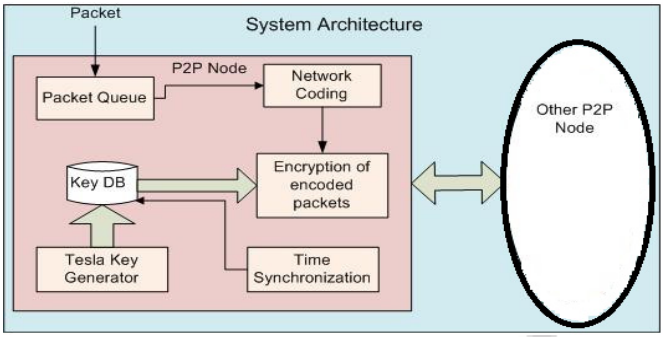


**Figure 1**: System architecture

- **Packet Queue**: This Queue contains the packet that is to be sent to the Next Peer.
- **Network Coding**: Encoding of the packets is done in this module.
- **Time Synchronization**: This module is responsible to maintain the time Synchronization between two peers while generating the key.
- **Tesla Key Generation**: The module where actually the key is generated for the encryption.
- **Key DB**: Stores the Key Generated by TESLA key generator.

## 6. Conclusions

Although the time keys scheme, which is a in-network security scheme based on time and space properties of network coding, is not a efficient scheme. As attacker can capture the times keys and may introduce an replay attack into the network. The proposed scheme use network coding along with the encryption of times keys. It saves the bandwidth of the network as well as it reduces total corrupted packet. This scheme is a complete defense mechanism for network coding-based P2P systems. It simultaneously provides both in-network detection and precise attacker identification for P2P systems. This detection and identification schemes are collusion resistant as well as tag-pollution resistant.

## References

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network Information Flow," IEEE Trans. Information vol. 46, no. 4, pp. 1204-1216, July 2000.
[2] "Ming He · Zhenghu Gong · Lin Chen · HongWang · Fan Dai · Zhihong Liu "Securing network coding against pollution attacks in P2P converged ubiquitous networks" Springer Science, Published online 26 June 2013.
[3] "The TESLA Broadcast Authentication Protocol" Adrian Perrig Ran Canetti J. D. Tygar
[4] Chi Cheng, Tao Jiang, Senior Member, IEEE, and Qian Zhang, Fellow, IEEE "TESLA-Based Homomorphic MAC for Authentication in P2P System for Live Streaming with Network Coding" published in IEEE Journal On Selected Areas In Communications/Supplement, Vol. 31, No. 9, September 2013