The Content Pollution in Peer-to-Peer Live Streaming Systems: Analysis and Implications

Sirui Yang¹, Hai Jin¹, Bo Li², Xiaofei Liao¹, Hong Yao¹, Xuping Tu¹

¹Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, China
hjin@hust.edu.cn

²Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China
bl@ust.hk

Abstract: There has been significant progress in the development and deployment of Peer-to-Peer (P2P) live video streaming systems. However, there has been little study on the security aspect in such systems. Our prior experiences in Anysee exhibit that existing systems are largely vulnerable to intermediate attacks, in which the content pollution is a common attack that can significantly reduce the content availability, and consequently impair the playback quality.

This paper carries out a formal analysis of content pollution and discusses its implications in P2P live video streaming systems. Specifically, we establish a probabilistic model to capture the progress of content pollution. We verify the model using a real implementation based on Anysee system; we evaluate the content pollution effect through extensive simulations. We demonstrate that (1) the number of polluted peers can grow exponentially, similar to random scanning worms. This is vital that with 1% polluters, the overall system can be compromised within minutes; (2) the effective bandwidth utilization can be sharply decreased due to the transmission of polluted packets; (3) Augmenting the number of polluters does not imply a faster progress of content pollution, in which the most influential factors are the peer degree and access bandwidth. We further examine several techniques and demonstrate that a hash-based signature scheme can be effective against the content pollution, in particular when being used during the initial phase.

1. Introduction

Peer-to-Peer (P2P) systems have gained great success in many popular Internet applications such as file sharing [1][2][3], Voice over IP (VoIP) [4] and live streaming services [5][6][7][8][9] due to its inherent scalability and easy deployment. However, these systems also allure attacks for its popularity and control of a large number of hosts.

The P2P epidemics [10][11] leverage the overlay networks while other intermediate attacks focus on the services themselves. Content pollution is a type of attacks that degrade the level of data availability by tampering the original content of targeted systems. This was closely examined for P2P file sharing [12][13]. To the best of our knowledge, there has been no systematical study on content pollution in P2P live streaming systems, yet the content availability is crucial for P2P live streaming system, which is the focus in this study.

Content pollution attackers participate in a streaming channel and compromise certain links. By tampering with video chunks, the attacker can disseminate the fake content with the help of the data exchange mechanism typically adopted in the P2P live streaming systems [5][7]. With the number of polluted chunks increases, users obtain useless chunks and consequently affect playback quality. Specifically, this results in two critical problems: first, the increase of interruptions during playback as the original content is unavailable after pollution; second, the reduction of peer sharing capacity. A former experiment [14] aiming at PPLive indicates that it is feasible to implement real content pollution in P2P live streaming systems, and the destructive attack compromised about 80% online peers in tens of minutes.

This problem is further challenged by the fact that the mechanism in handling content pollution in P2P file sharing can not be directly applied in P2P live streaming. In file sharing, the contaminated files or chunks can be either recovered or deleted that greatly helps to reduce the pervasion [13][15]; while in live streaming, it is difficult to detect and recover such chunks in the often small playback window without violating the playback deadlines. In addition, the existing content availability exchange schemes mostly rely on the Gossip protocol, which does not have effective mechanism to deal with the contaminated content. Therefore, the traditional countermeasures such as blacklisting [16], hash-based signature [17] and data encryption [18], will either increase the communication overhead or delay the video playback.
In this paper, we examine the content pollution for P2P live streaming systems. Derived from random scanning epidemics [19], we propose a probabilistic model to capture the progress of content pollution. We verify the proposed model using a prototype based on a real system, Anysee [7] [8], which we have implemented. Further, we carry out extensive simulations to study the impact on various performance aspects of the system from the content pollution. Our major contributions in this study are: (1) to the best of our knowledge, this is the first attempt in presenting a formal model for the content pollution in P2P live streaming systems. The model is verified from a prototype based on Anysee and through extensive simulations; (2) our study conclusively demonstrates that the most crucial factor in content pollution is not the number of polluters, but are the peer degree and access bandwidth; (3) we present concrete evidences in that traditional countermeasures including blacklisting, flow encryption and hash-based chunk signature are insufficient against the content pollution attack; finally (4) we show that the hash-based signature solution taking effect in the initial phase of the pollution process is most effective among all the countermeasures.

The rest of this paper is organized as follows: in Sec 2, we discuss the related works. The basic system architecture is introduced in Sec 3. In Sec 4, we propose the probabilistic model. The simulation and the analysis are presented in Sec 5. In Sec 6, we discuss the implications of countermeasures for content pollution. In Sec 7, we conclude the paper with highlights on the further research.

2. Related Work

Content pollution is prevalent in P2P file sharing because of the copyright infringement problem [12] [20]. Liang et al. [12] presented a measurement study on content pollution in KaZaA/FastTrack networks [2], which showed that the pollution was pervasive for recent popular songs and up to 50% versions were polluted. Kumar et al. [13] developed fluid models for pollution proliferation that captured a variety of user behaviors including propensity for popular versions and freeloading. Christen et al. [20], similar to the work in [12], illustrated that a potentially strong correlation existed between content availability and topological properties of the underlying P2P network.

Dhungel et al. [14] carried out a pollution experiment for PPLive system [9], which is one of the largest P2P live streaming systems. In the experiment, two peers in Brooklyn and Hong Kong were used respectively as sampling peers to gather pollution information. It was observed that the video playback quality could be significantly affected with the pollution.

Hardarison et al. [21] studied the application-level multicast system and discussed four types of attacks: membership attack, forgery, denial-of-service (DoS) and omission attack. They also proposed and evaluated an intrusion-tolerant system, called Secure-Stream. Wang et al. [22] investigated the DoS attacks in P2P streaming and proposed Ripple-Stream to safeguard the system from such an attack.

Our study is different from all prior works in that: 1) we present the first system model that captures the effect of content pollution in P2P live streaming systems; 2) we identify the critical factors in content pollution; and 3) we illustrate the mechanism that can be effective against content pollution.

3. System Architecture

In this section, we describe the basic mechanism in the P2P live streaming system and illustrate the scheme for content pollution attack. We use Anysee [7] [8] as the base system, which resembles the typical P2P live streaming systems similar to Coolstreaming [5] [6] and PPLive [9]. There are two distinctive modules in such systems: topology formulation and data exchange.

**Topology formulation:** Anysee organizes peers into an unstructured random topology, in which each peer maintains connections to a number of other peers or “neighbors”. When a peer joins the system, a bootstrap server that maintains the indices of online peers returns a group of peers to act as its neighbors. In Anysee, neighbors periodically exchange “heartbeat” messages that contain the information of the sender’s neighbor list. Therefore, a peer node can select new neighboring peer(s) from such “candidate neighbors” whenever the current neighbors become insufficient for streaming requirement.

**Data exchange:** The content pollution is most relevant to the data exchange mechanism. In Anysee, the content organized in chunks is provided by a source server. Each chunk has a timestamp indicating its playback time. Each peer maintains a buffer with the size of 160 seconds. According to the different playback deadlines, the first 10% chunks are categorized as “urgent data”, the last 10% chunks are “ease data”, and the middle 80% chunks are called “common data”. Peers use a Gossip protocol to periodically exchange their content availability information in the buffers with the neighbors using the “buffer map message”. A buffer map is a bit-wise representation of the chunk availability, i.e., if a chunk is available, the corresponding bit is set to “1”, otherwise “0”. On receiving a buffer map message, a peer by comparing with the local buffer map can determine the chunks that can be potentially swapped with neighbors. Anysee employs a bandwidth estimation mechanism to evaluate the bandwidth of all neighbors, which provides diversity when requesting data from different neighbors.
Content pollution: we envision and design a simple content pollution in Anysee as follows: first, an attacker downloads the Anysee software and designs a hook program (in Windows OS) to act as an Anysee client. This allows the hooked client to build up the neighbor list with other normal peers. Afterwards, the hook program continuously tampers with the buffer or packets which are provided to neighbors. Eventually, the neighbors get polluted chunks and recursively disseminate them to the whole network.

4. The Content Pollution Model

Based on the above mechanism, we propose a probabilistic model to capture the progress of content pollution in the P2P live streaming system.

4.1. The General Model

We define that a peer is a “polluted peer” if it has received at least one polluted chunk and a “clean peer” otherwise. So a peer can be either in “polluted state” or “clean state”; by definition the polluter itself is a polluted peer. We denote $x$ as the number of clean peers and $y$ for the polluted peers in the system. We can then define $(x, y)$ as a Markov process with the state transitions as follow:

- $(x-1, y+1)$: when a clean peer gets a polluted chunk;
- $(x, y)$: otherwise.

This can be solved by a similar technique used for random scanning worms [19], which determines next victims by randomly selecting a series of IP addresses:

$$
\alpha = \frac{e^{s(t-\Gamma)}}{1 + e^{s(t-\Gamma)}}
$$

(1)

Wherein $\alpha$ is the proportion of infected hosts at time $t$, $v$ is the vulnerability density (the proportion of addresses that are vulnerable), $s$ is the scan rate (the number of random probes during each unit of time a worm instance performs) and $\Gamma$ is a threshold. In P2P live video streaming, $v=1$. Because the pollution process is solely based on chunk transmission influenced by the minimum of the neighbor number and the bandwidth among neighbors, $s$ is also limited by this factor. Specifically, let $c$ denote the total upload bandwidth of a peer (represented in chunks per second), $k$ denote the number of its neighbors and $\rho$ represent the proportion of polluted peers at a given time $t$. Thus, we can obtain a general model:

$$
\rho = \frac{e^{\theta \min(c, k)(t-\Gamma)}}{1 + e^{\theta \min(c, k)(t-\Gamma)}}
$$

(2)

Where $\theta$ is a polluting rate, which needs to be determined. Note that in practical, the media content in one second is always divided into several chunks for transmission. However, for simplicity, we assume a chunk contains one second media content in following discussion.

4.2. Determine $\theta$

The key in the general model (2) is how to determine the parameter $\theta$. Without losing the generality, we assume the peers exchange content synchronously. This simplification is acceptable, as the timing difference among peers does not influence the pollution process. We also assume the bandwidth and the degree are even for all peers as their difference has been considered in the general model.

Our methodology is: first, to investigate the newly polluted peers in each round; and find the relation between the sum of polluted peers and time; finally, get parameter $q$ with differential operations. The notations are listed in Table 1.

<table>
<thead>
<tr>
<th>$AR$</th>
<th>number of newly polluted peers</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>sum of polluted peers</td>
</tr>
<tr>
<td>$k_i$</td>
<td>number of neighbors of Peer $i$</td>
</tr>
<tr>
<td>$r$</td>
<td>polluted chunks in polluter’s buffer</td>
</tr>
<tr>
<td>$s$</td>
<td>Buffer size (in chunks)</td>
</tr>
<tr>
<td>$L$</td>
<td>gap of playback time between neighbors</td>
</tr>
<tr>
<td>$t_0$</td>
<td>starting time of pollution</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Poisson rate of polluter joining process</td>
</tr>
<tr>
<td>$T$</td>
<td>period of time for each round</td>
</tr>
<tr>
<td>$\delta(t)$</td>
<td>number of peers receiving multiple polluted chunks at time $t$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>probability for polluters being neighbors</td>
</tr>
<tr>
<td>$N$</td>
<td>network size</td>
</tr>
<tr>
<td>$K$</td>
<td>number of polluters at the beginning</td>
</tr>
</tbody>
</table>

Single polluter: First, we consider a simple case where there is only one polluter. The distribution of chunks among peers in a random topology can be regarded to be random. During each round, the pollution will proceed only between the polluter (or a polluted peer) and its neighbors. So the probability ($p_c$) for a neighbor to request a polluted chunk from the polluting peer is

$$
p_c = \frac{L}{s} \times \theta
$$

(3)

The $\theta$ is defined as the ratio of requested chunks from the polluting peer. Suppose that the polluter has chunks from $[1’30’’, 4’10’’]$ and the neighbor’s is from $[2’20’’, 5’00’’]$, then requested chunks are $[1’30’’, 2’19’’]$. The length of the request chunks is 50 seconds. If $s=160$ (seconds) as set in Anysee, we can get $\theta=50/160=0.3125$ in this case. Further, if the polluter tampers a half of the content, i.e., $r=80$, then $p_c=0.156$.

By accumulating all the probabilities of the $s$ chunks, we get the requested chunks between a polluting peer and all its neighbors:

$$
\theta = \sum_{i=1}^{s} \left( C^i_r \cdot \frac{1}{L} \right)
$$

(4)

The $C$ is the mathematical combination operator. Thus, the probability ($p$) for the $i^{th}$ neighbor of the polluter being polluted in one round is

$$
p = p_c \frac{1}{n_{ci}} \sum_{i=1}^{s} \left( C^i_r \cdot \frac{1}{L} \right)
$$

(5)
Since the polluter has \( k_i \) neighbors, the number of newly polluted peers during each round is

\[
\Delta R = k_i \cdot p = \frac{r}{s} \cdot \frac{k_i}{k} \sum_{i=1}^{s} (C_i^r \cdot \left(\frac{1}{L}\right)^i)
\]  

(6)

Given each peer degree is the same as assumed, i.e., \( k_i = k \), we have,

\[
\Delta R = \frac{r}{s} \sum_{i=1}^{s} (C_i^r \cdot \left(\frac{1}{L}\right)^i)
\]  

(7)

We assume the relation between the newly polluted peers and the sum of ever polluted peers follows the Bank Interest Law. So, after the \( m \) rounds of chunk exchange, the sum of polluted peers (\( R \)) is

\[
R = \begin{cases} 
1, & \text{if } m = 0 \\
(\Delta R + 1)(1 + \frac{\Delta R - \delta(m)}{\Delta R + 1})^{m-1}, & \text{if } m \geq 1
\end{cases}
\]  

(8)

**Multiple polluters:** We now extend the model with multiple polluters. A simple case would be that \( K \) polluters are evenly distributed in the network and pollutions occur at the same time. Then the effect of content pollution is simply \( K \) times of the one in the single-polluter case

\[
R = \begin{cases} 
K, & \text{if } m = 0 \\
K(\Delta R + K)(1 + \frac{\Delta R - \delta(m)}{\Delta R + 1})^{m-1}, & \text{if } m > 1
\end{cases}
\]  

(9)

A more realistic case is that the \( K \) polluters launch their attacks at different times. We further assume that this follows an independent Poisson process with parameter \( \lambda \). Polluters can be neighbors with the probability \( \mu \). The number of polluted peers after all polluters’ attacks is:

\[
R = (\Delta R + \lambda(t-t_0) - \mu)(1 + \frac{\Delta R - \delta(t)}{\Delta R + \lambda(t-t_0) - \mu})^{\frac{t-t_0}{T}}
\]  

(10)

Where \( t_0 \) is the time for the first polluter to start attack and \( \frac{t-t_0}{T} \approx m - 1 \).

The variable \( q \) in (2) specifies the polluting rate, thus \( q = \frac{\partial R}{\partial m} \). We next define a few notations to simplify the formula in above expressions:

\[
\phi = K(\Delta R + K)
\]  

(11)

and

\[
\psi = 1 + \frac{\Delta R - \delta(t)}{\Delta R + 1}
\]  

(12)

With some basic differential computations, we have

\[
q = \frac{\partial R}{\partial m} = \phi \cdot \psi^{-1} \cdot \ln \psi
\]  

(13)

### 4.3. The Refined Model

In this subsection, we combine aforementioned formulas to derive the refined probabilistic model for content pollution in P2P live streaming.

By replacing \( m-1 \) with \( \frac{t-t_0}{T} \), we integrate (2) and (13) and we obtain the refined probabilistic model.

\[
\rho = \frac{e^{\phi \psi^{\left(\frac{t}{T}\right)}} \cdot \ln \psi \cdot \min(c, k)(t-\Gamma)}{1 + e^{\phi \psi^{\left(\frac{t}{T}\right)}} \cdot \ln \psi \cdot \min(c, k)(t-\Gamma)}
\]  

(14)

We notice the progress of content pollution is affected by following factors. (1) The time to start the pollution attack: This time will partly influence \( L \). When \( L \) is smaller, the probability of a polluter to hold a chunk which is desired by neighbors gets larger. It is accordant to the fact that if a polluter joins the channel earlier than its neighbors, it is possible to provide them more chunks; (2) Initial polluted pieces of video chunks (\( c \)). If a polluting peer has all its possessed chunks contaminated, the efficiency is highest. (3) Number of peers received multiple polluted chunks (\( k(t) \)). The variable is getting large as more peers become compromised. We simply regard this factor as a monotonically increasing function over time. The slope of the increment is however slowed down with the polluted peers increase. Finally, it reaches to a constant value if the topology is stable. We do not analyze this function further because it is influenced by the network churn and user interaction. (4) Times of polluters to be neighbors (\( m \)). If polluters are neighbors or share neighbors, the overall effect is a bit decreased. However, if the attacker deploys polluting hosts well enough, this variable could be zero.

### 5. Simulation and Evaluations

In this section, we verify the model with simulation and examine the impact from content pollution.

#### 5.1 Methodology and Parameters

The evaluation is carried out in two phases: (1) model validation and attack effects: we first simulate a P2P based live streaming network where 1% polluters are injected. We observe the polluted effects such as the polluted peer number and wasted bandwidth; (2) parameter influence: we vary the size of the network, the degree of peers and the distribution of peers’ access bandwidth, as well as the proportion of polluters. Then we observe and compare the corresponding results to discover the most influential factors during content pollution.

The simulation is set up resembling the Anysee system that we developed and has been in use since 2004. The basic parameters are summarized in Table 2. Note that earlier Anysee versions use \( s = \) 80 seconds but update it to \( 160 \) seconds in the latest version. For simulation time reduction consideration, we still use \( s = 80 \) in simulations.

**Link settings:** We take the real measurement results from P2P networks [1] to set the underlying links for each peer: (1) Latency: it is observed that latencies of 20% nodes in such P2P networks are less than 70ms, another
20% are larger than 280ms, and the remained 60% are between the two. (2) Access bandwidth: roughly 80% nodes access the P2P network with ADSL and 20% with a modem. The download and upload bandwidth pair are correspondingly set to (1024, 512) and (56, 14) kbps. We simulate a P2P live streaming network with up to 5000 peers and the bitrate is 400 kbps.

**Model settings:** The parameter values in simulation are specified in Table 2. The polluters join the channel after about \( L = 80\text{~to~}110 \) seconds after the simulation starts. This ensures that the gap of playback time between polluters and their neighbors does not deviate much from real occasions. As we discussed in Sec 4.3, the polluters tamper all its buffered content (i.e., \( r = 160 \)) and the adversary deploys polluters dispersedly (i.e., \( \mu = 0 \)).

### 5.2 The Performance Metrics

We use four metrics to investigate the progress and effect of content pollution: (1) number of polluted peers and number of clean peers; (2) system down time, which captures the elapsed time for all peers being polluted in networks. It illustrates the efficiency of the content pollution attack; (3) wasted bandwidth. The propagation times of polluted chunks (both for a single peer and for the whole network) are recorded to represent the wasted bandwidth. Similarly, we calculate the propagation times of clean chunks for comparison; (4) average loss of chunks. As the playback QoS perceived by a user is difficult to quantify, we use the average loss rate of original chunks from each peer as an indicator. The loss rate is the proportion for polluted chunks in the buffer.

### 5.3 Results and Analysis

We present and analyze the simulation results in this subsection. Note that the origins of the legend are mostly from 80 second when the pollution is issued.

#### 5.3.1 Model Verification and Attack Effects

**Model verification:** The progress of pollution in networks with different size is shown in Figure 1. The growth of polluted peers over time is similar in all curves. Therefore, we take the 3000-peer network curve for illustration to verify the probabilistic model. Above all, we denote the three phases in the progress: (1) the initial phase where the curve is slowly rising near the origin point; (2) the propagation phase where the curve rises sharply; (3) the tail phase where the curve slows down.

We extract the 3000-peer network result and compare it with the model in Figure 2. We first determine the shapes of the two curves are similar. Then we find out that when \( T = 30 \) (seconds) and the value of the power is 0.3 in Formula 14, the theoretical curve is closest to the experimental one. The overlap of the experimental curve and the theoretical one indicates that the proposed model can describe the progress of content pollution.

It further verifies that the random scanning worms and the content pollution have essentially same characteristics. Since there are few effective solutions to random scanning worm, it also implies why the content pollution attack is difficult to be eliminated.

**System down time:** In Figure 1, all the peers have been polluted in less than 60 seconds (i.e., from 80 second to 140 second). Thus, the system down time is about 1 minute. It is also indicated that the time is independent of the network size.

Comparing our simulation results to the experimental results in [14], it is noticed that the pollution rate observed in real systems is slower than in the simulation. The former takes tens of minutes, while ours is about one minutes. We estimate the reasons for this difference are: (1) full connection and abundant access bandwidth facilitate the attack in the simulation; (2) in real systems, polluted peers...
may leave the system at random; (3) the number of polluters is much less in [14] (only one or two).

**Number of polluted peers:** This metric can be easily obtained from Figure 1: 20% peers have been polluted within about 10 seconds in the initial phase; 80% peers are compromised within 20 seconds during the propagation phase; finally, all peers are polluted at about 60 seconds. The tail phase is almost twice of the other two. The implication of the three phase growth of polluted peer number will be further discussed in Sec 6.

**Wasted bandwidth:** In Figure 3, the sum of packet transferred among peers during the pollution period in the 3000-peer network is presented, including both the original (clean) packets and polluted (dirty) packets. From the figure, it is clear that although the initial number of dirty packets is extremely small, it becomes significant along with the increases of the polluted peers. For example, at 140 second (when all peers have just been polluted as shown in Figure 1) $2.0 \times 10^5$ dirty packets and $1.5 \times 10^6$ clean packets have been transferred. Therefore, about 11.8% of bandwidth is wasted. This is further aggravated along the time, since more peers are polluted. For clarity, we record the increment of the transferred packets during each second, as shown in Figure 4. This figure also confirms that with more peers polluted, the increment of dirty packets is more significant. After the whole network is compromised (140s~160s), the transferred dirty packets in each second usually overwhelm the clean ones.

**Average loss of chunks:** The number of polluted chunks in a peer’s buffer directly affects the degree of interruption in playback. The average loss of chunks in the buffer for each peer during the pollution process is shown in Figure 5 for the 1000-peer and 3000-peer networks, respectively.

From the figure, it is observed that up to 17 chunks in each second are polluted in the 1000-peer network. As the length of buffer is 160, the loss ratio of chunks is 10.6%. Correspondingly, the ratio reaches 11.2% in the 3000-peer network. A chunk contains the content in a second. This affects viewing experience substantially.

**5.3.2 Influence of Parameters**

We investigate the influence of polluter numbers ($K$), start time of pollution ($L$), peer bandwidth ($c$) and average peer degree ($k$) in this subsection.

**The effect of polluter number:** We choose 0.1%, 1% and 10% peers as polluters and the results are shown in Figure 6. We can see that the difference is not significant even the number of polluters is significantly different. The reason is that the content pollution is devastating because all peers upon reception of polluted content will participate in the progress. In another word, with one or very polluters, the system can be easily compromised.
The effect of pollution start time: Figure 7 is the result with different pollution start time. Although earlier pollution start time means earlier system down time, the lengths of the time used to crash the whole system are nearly the same (about 1 minute).

The effect of bandwidth: We adjust the percentages of ADSL and modem users in the simulation and thus change the average bandwidth of peers. Figure 8 shows the result. Intuitively with more bandwidth, the pollution process is faster since more dirty chunks can be spread. With no ADSL peers, the average bandwidth is poor and the pollution needs more than 160 seconds to compromise the whole network, while this period is shortened to 10 seconds if the percentage of ADSL peers rises to 100%. Further, the progress of pollution is exponential to the percentage of ADSL peers. It indicates that in broadband accessing regions, pollution boasts a more destructive power than in bandwidth deficient regions.

The effect of average peer degree: We change the average peer degree from 10 to 80, and the result is shown in Figure 9. The result indicates that a polluter can affect more neighbors when the peer degree is larger. Thus, the progress of pollution is faster. From the figure, we observe that the average peer degree also exponentially influences the pollution progress. This is confirmed by the form of our model where the peer degree \( k \) plays an equivalent role as the bandwidth \( c \) does.

6. Implications

More effects of content pollution: Besides the impact on the playback interruption and wasting available bandwidth, there are some other negative effects to sabotage the targeted system by content pollution attacks.

Most existing P2P live streaming systems encourage peers to transfer “precious” chunks urgently. There are two kinds of precious chunks in fact: the one with the earliest timestamp and the one with the most recent timestamp. The former one is most needed by new-arriving peers, while the latter requires being propagated as soon as possible. Then the attacker can modify the timestamp to accelerate the transmission of the polluted chunks.

Weakness of traditional solutions: There are mainly three kinds of traditional countermeasures for content attacks: blacklisting, flow encryption and chunk signature.

Blacklisting [16] is not effective in P2P streaming. On one hand, there is no explicit concept of “identification” in existing applications. Peers can only be recognized by their IP addresses, which is easy to cheat. On the other hand, even if the Global Unique Identity (GUID) concept is introduced in such systems, the polluter can simply re-validate itself by registering a new identity. Further, it is difficult to identify whether a peer is a polluter or a victim of pollution by blacklisting.

The Public Key Infrastructure (PKI) uses encryption to prevent the flow from modification [18], which is not suitable in P2P environment. Partly because it is difficult to deploy appropriate trusted third parties in such networks [18]. In addition, the implementation of an infrastructure and encryption is either incompatible or costly with the existing P2P live streaming systems.

A lightweight measure is the hash-based chunk signature scheme. In hash-based signature schemes, the hash value of the chunk (or a segment of the chunk) is computed and rides on the chunk. A peer will check this value before using and spreading the chunk. The countermeasure is able to avoid polluted content being used in a certain degree. Our simulation shows that the periods of initial phase and propagation phase are of similar length, which implies measures taken in the initial phase can be
much more effective. The hash-based scheme takes effect in the initial phase, so it is able to discover polluters in an early period.

7. Conclusion and Future Work
In this paper, we have studied the content pollution attack in P2P live streaming systems. We establish a probabilistic model for content pollution. The results show that the content pollution is destructive to such systems. We observe from our model and simulations that the most influential factors for content pollution are the access bandwidth and the degree of participating peers, not the number of initial polluters. Also we argue that the hash-based solution can be potentially effective against such attacks, esp. when being deployed during the initial stage.

We believe much remains to be done. We are currently expanding our works to consider: (1) the design of a lightweight countermeasure mechanism, both flexible and effective for content pollution in P2P live streaming; (2) the impact of content pollution in more structured based P2P live streaming systems.

8. References