Signature for Content Distribution with Network Coding

Fang Zhao LIDS, MIT

Ton Kalker HP Labs

Muriel Médard LIDS, MIT

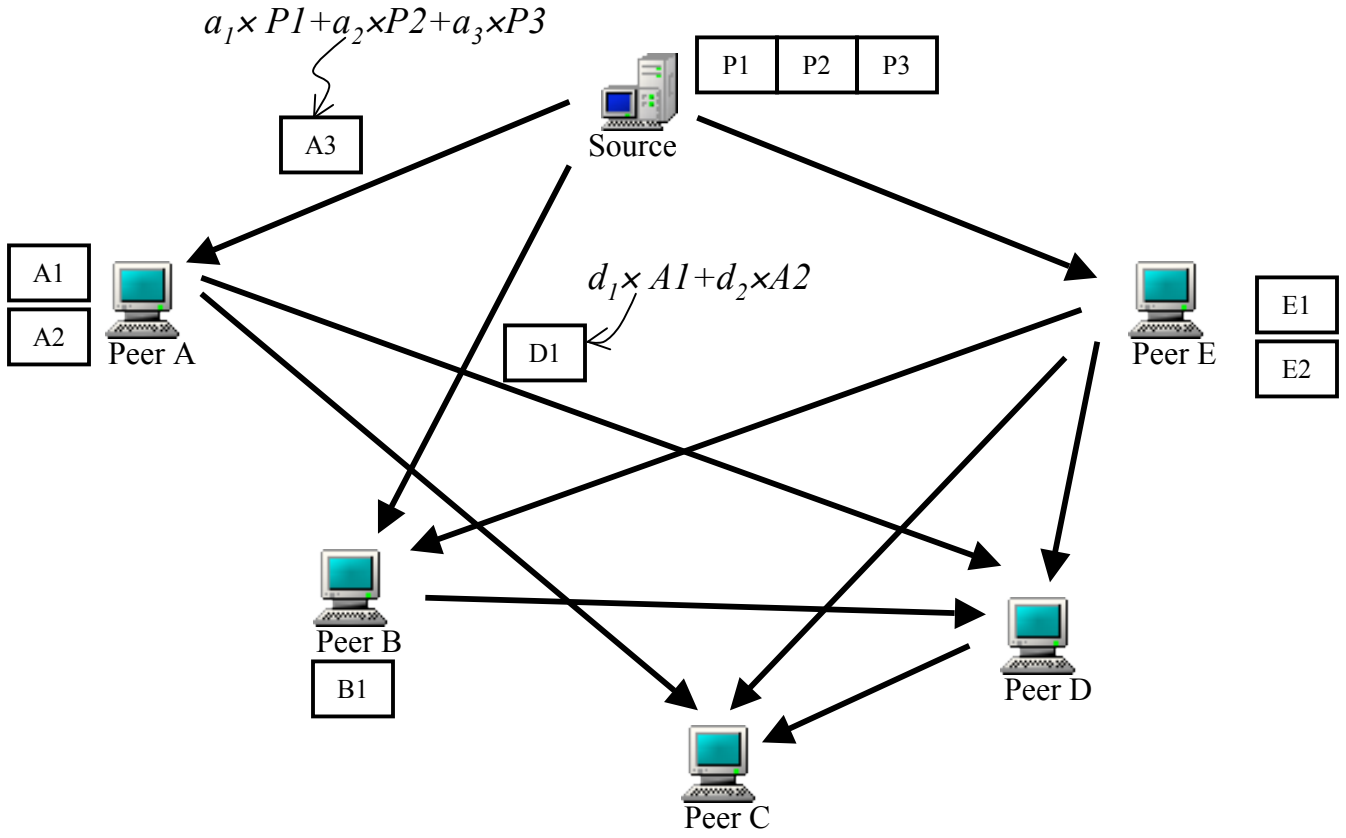
Content distribution of large files

- Distribution of large files to many users.
- Traditional solutions are based on a client-server model.
- Alternative technique - P2P swamping.
- Example - BitTorrent
  - Divide file into many pieces.
  - Client requests different pieces from server(s) or other users.
  - Client becomes server to pieces downloaded.
  - When a client has obtained all pieces, re-construct the whole file.
  - Problem: hard to do optimal scheduling of pieces to nodes.

Content distribution using network coding

- Use network coding to increase the efficiency of network coding in a P2P cooperative architecture.
- Instead of storing pieces on servers, store random linear combination of the pieces on servers.
- Clients also generate random linear combination of the pieces they have received to send out.
- When a client has accumulated enough degrees of freedom, decode to obtain the whole file.

Content distribution using network coding



Security for network coding

- Network coding is vulnerable to pollution attacks by malicious nodes in the network.
- Malicious user can send packets with valid linear combination in the header, but garbage in the payload.
- The pollution of packets spreads quickly.
- Need a homomorphic signature scheme that allows nodes to verify any linear combination of pieces without contacting the original sender.

Problem formulation

- A source *s* wishes to send a large file to a group of peers, *T*.
- View the data to be transmitted as vectors $\overline{\mathbf{v}}_1,\ldots,\overline{\mathbf{v}}_m$ in *n*-dimensional vector space F_p^n, where *p* is a prime. The source node augments these vector to $\mathbf{v}_1,\ldots,\mathbf{v}_m$ given by

$$\mathbf{v}_i = (0,\ldots,1,\ldots,0,\overline{v}_{i1},\ldots,\overline{v}_{in})$$

  where the first *m* elements are zero except the *i*-th one is 1, and $\overline{v}_{ij} \in F_p$.
- Each packets received by a peer is a linear combination of all the pieces.

$$\mathbf{w} = \sum_{i=1}^{m} \beta_i \mathbf{v}_i$$

Signature for network coding

- The vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m$ span a subspace ***V*** of F_p^{m+n}.
- A received packet is a valid linear combination if and only if it belongs to ***V***.
- Each node verifies the integrity of a received vector ***w*** by checking the membership of ***w*** in ***V***.
- Our approach has the following ingredients:
  - *q*: a large prime such that *p* is a divisor of *q* -1.
  - *g*: a generator of the group *G* of order *p* in F_q .
  - Private key: $K_{pr} = \{a_i\}_{i=1,\ldots,m+n}$, a random set of elements in F_q^* .
  - Public key: $K_{pu} = \{h_i = g^{a_i}\}_{i=1,\ldots,m+n}$.

Signature for network coding

- The scheme works as follows:
  - The source finds a vector ***u*** that is orthogonal to all vectors in ***V***.
  - The source computes vector $\mathbf{x} = (u_1 / a_1, \ldots, u_{m+n} / a_{m+n})$.
  - The source signs ***x*** with some standard signature scheme and publishes it.
  - When a node receives a vector ***w*** and wants to verify that ***w*** is in ***V***, it computes

$$d = \prod_{i=1}^{m+n} h_i^{x_i w_i}$$

and verifies that d =1.

Discussion

- It can be shown that it is as hard as the $(p, m, m+n)$ Diffie-Hellman problem
- Thus, it is as hard as the Discrete Logarithm problem to find new vectors that also satisfy the verification criterion other than those that are in $\boldsymbol{V}$
- Overheads
  - Part of the public key $\boldsymbol{K}_{pu}$ has to be re-generated for each file, otherwise a malicious node can use the information from the previous file
  - Signature vector, $\boldsymbol{x}$

Discussion

- If the file sizes are large, after the initial setup, each additional file distributed only incurs a negligible amount of overhead using our signature scheme.
- Under our assumptions that
  - there is no secure side-channel to transfer hash values from the source to all the peer nodes, and;
  - all peers have full knowledge of the public information of the security scheme,

  our signature scheme has to be applied on the original file, not on hashes.

Conclusions

- Proposed a solution to the security problem in content distribution with network coding.
- Use a signature vector for each file that can be used to easily check the integrity of all the packets received for this file.
- This scheme is secure and has low overhead.