Open access • Proceedings Article • DOI:10.23919/SOFTCOM.2017.8115565

# Towards authenticated network coding for named data networking — Source link

Ryma Boussaha, Yacine Challal, Malika Bessedik, Abdelmadjid Bouabdallah

**Institutions:** École Normale Supérieure, University of Technology of Compiègne

**Published on:** 01 Sep 2017 - International Conference on Software, Telecommunications and Computer Networks

**Topics:** Linear network coding, Encryption, Network packet, Homomorphic encryption and Cryptography

# Towards Authenticated Network Coding for Named Data Networking

Ryma Boussaha, Yacine Challal, Malika Bessedik, Abdelmadjid Bouabdallah

# Towards Authenticated Network Coding for Named Data Networking

Ryma BOUSSAHA*, Yacine CHALLAL*§, Malika BESSEDIK* and Abdelmadjid BOUABDALLAH§
*Laboratoire LMCS, Ecole nationale Supérieure d'Informatique , Algiers, Algeria
§Laboratoire HEUDIASYC, UMR CNRS 6599, Universite de technologie de Compiègne, France
Email: {r_boussaha, y_challal, m_bessedik}@esi.dz, bouabdal@utc.fr

***Abstract*—Named Data Networking represents a novel and an alternative approach to the current host based Internet architecture, in which the content becomes the core of the communication model. In this paper, we propose to improve named data networking robustness and throughput performances by introducing network coding functionalities. We also propose an optimized authentication model based on homomorphic encryption to deal with the "pollution attacks" problem in which malicious nodes can flood the network with bad packets and prevent the reconstruction of information at recipients. We first present our optimized homomorphic signature scheme with respect to the practical constraints such as the processing overhead generated by complex cryptographic operations and we define the network coding scheme tailored for named data networking. Our numerical results demonstrate that the use of this optimized model reduces by 10% the overall calculation cost induced by cryptographic operations compared to a hop-by-hop coding and verification model.**

***Index Terms*—Named Data Networking, Network Coding, Pollution Attack, Homomorphic Encryption.**

## I. Introduction

The Internet architecture today is strongly driven by the naming of content objects instead of addressing end-hosts. The Palo Alto Research Center (PARC) describes a potential new architecture, named Content Centric Neworking (CCN), which deals with named content, rather than its physical location. This proposal changes radically data transfer by pushing content storage and delivery at network layer itself. Named Data Networking (NDN) [1] is a prominent collaborative research effort that depicts the content centric approach to networking. The NDN paradigm suggests to grant universal in-network caching, in a manner that the closest node in the network can potentially serve the solicited content as efficiently as possible without connecting to the origin server.

In named data networking, a content is divided into small sized chunks to enhance the caching efficiency, and the Internet bandwidth is withered by repeated downloads of popular content. To fully exploit the parallel transmission and the redundant chunk sources, we propose in this paper to introduce a network coding [2] method which brings significant benefits into networking, namely enhanced throughput and reduced delivery time. With network coding, sources and intermediate nodes randomly combine the content chunks with each other and transmit the resulting linear combinations to their adjacent nodes, while allowing the final recipients to obtain the original information. The reason for using network coding in NDN applications are twofold. First of all, NDN-NC[1] consent to fully exploit network coding in order to improve transmission efficiency and scalability, as well as resilience to attacks. This is also used to reach the maximum possible information flow in the network. Secondly, NDN-NC takes advantage of caching functionality of named data networking to improve network robustness. NDN leverages in-network caching for further use. Each node can cache forwarded content, with this cache mechanism content will be quickly spread in the network. However, with NDN the consumer uses only the first coming back content, and any subsequent copies will be discarded. Furthermore, unnecessarily extra traffic introduced by these redundant copies returned to the consumer will overwhelm network resource. To fully utilize these redundant data sources, this paper brings network coding and NDN together to leverage redundant packets to carry out useful coded information, so that recipients receive the required linear combinations of packets. On the other hand, NDN-NC can improve robustness of network. If a message is lost on some path due to path failure or congestion, it could be easily recovered with a simple linear combination holding it.

Although network coding has gained significant attention by improving resilience to random packet loss and by increasing throughput and reliability, it is highly sensitive to "pollution attacks" [3]. If some nodes are malicious and inject invalid linear combinations of received data, then even a single faulty packet is likely to contaminate the whole network and eventually prevent the reconstruction of information at targets. To solve this issue, it is desirable to have a "hop-by-hop containment". However, standard data origin authentication techniques based on digital signatures or MAC cannot mitigate pollution attacks since recipients do not have the original message packets and therefore cannot verify the proof of authenticity. Prior work has shown that network coding signature schemes relying on homomorphic hash function [4] or homomorphic signatures [5] can be used to solve this problem. These primitives are designed in a way that a signature (or a hash value) of a linear combination of vectors results in a corresponding homomorphic combination of signatures (or hashing). Nonetheless, the practical aspects of secure network coding implementation in a content centric networking architecture have not been well investigated in previous work. These schemes generate

[1]Named Data Networking with Network Coding.

an important computational overhead since they need complex cryptographic computations like modular operations, exponentiations, multiplications, etc.

In fact, the initial attempts towards combining ICN and network coding are only at their early stage. In [6], the authors investigate the required architectural changes that arise from the semantic difference between naming. In [7], the authors propose a novel signature scheme compatible with network coding, they propose also a forwarding plane to observe the network state, such as network failure, link transmission performance and distribution of coded packets. In [8], the authors design a special Interest coding and forwarding strategies for getting linearly independant coded blocks simultaneously from multiple nodes. However, it is unclear to what extent the network architecture should be modified to incorporate fully homomorphic encryption scheme to ensure both authentication and integrity towards pollution attacks.

We contribute to this area in several ways. First, we assess the practical feasibility of this homomorphic encryption scheme and we formulate the multiobjective homomorphic signature and network coding assignment model considering practical constraints such as the processing overhead generated by complex cryptographic operations. Second, we propose a network coding scheme tailored for named data networking. We define the corresponding Interest and data forwarding strategies. Our solution achieves the best trade-off between communication overhead and computation overhead while ensuring packets authentication.

The remainder of this paper is organized as follows. In Section II, we describe our optimization coding and homomorphic security model for network coding system and we present the construction satisfying our network architecture. The performance of the proposed security scheme are shown in Section III and Section IV concludes the paper and gives some directions for future work.

## II. Construction of network coding model in NDN

In our proposed architecture NDN-Auth[2], we assume that the network is represented with a directed graph $G = (V, E)$, such that $V$ is the set of vertices $(nodes)$ and $E$ is the set of edges $(links\ between\ nodes)$. The set $V$ is partitioned into three subsets, $V = C \cup P \cup I$, where $C$ is the set of consumers which send Interest messages, $P$ is the set of producers which respond with Data messages and $I$ is the set of intermediate nodes which support the forwarding of Interest/Data messages and store the received chunks.

Assume a client node $C$ intends to request a file $F$ from a producer $P$. Then P (or the encoder) divides the file $F$ into $m$ chunks (or packets) $p_1, p_2, p_3, ..., p_m$ where the size of each packet is $n$, and creates augmented vectors[3] from these packets. Thereafter, the encoder generates and transmits linear combinations of these vectors. The intermediate nodes, called also recoders receive some series of encoded packets and perform new linear combinations that are passed to the decoder which will be able to reconstruct the original file after receiving at least m linearly independant packets. In order to ensure authentication, the source and the set of intermediate nodes must append fully homomorphic signatures to the transmitted coded messages. While homomorphic methods, especially those based on public key cryptography, provide strong protection against both external and internal attacks, they do increase processing overhead. In this paper, we use the more practical variant of the homomorphic encryption scheme recently proposed in [9].

In this section we present the architectural design of our proposed model combining network coding with named data networking. Firstly, a slight hop-by-hop containment is performed to promote homomorphic signature scheme calculation cost. We assume that compelling nodes are installed in the network, these nodes should perform encoding, signature verification and creation of new homomorphic signatures operations. Otherwise, the other nodes should only transfer packets to the requesting interfaces. This method will make it possible to reduce the overall calculation cost and ensure a pact between it and security. The selection of trust nodes is ensured by means of a multiobjective optimization model. Secondly, an adjustment of the network coding approach to named data networking is ensured by the proposal of an interest forwarding strategy and a data forwarding strategy.

### A. Multiobjective optimization model

In this paper, we study the optimal coding assignment with practical constraints such as the coding and signature cost, the decoding and verification cost and the required security level. We consider the case of single producer [4] and single consumer nodes. As described above, the network can be modeled as a directed graph. We formulate a multiobjective mixed integer program (MIP) to find the optimal coding nodes assignment and to determine the number of linear combinations operated by each one. We assume that the required security margin is given. We consider that the content is a complete file consisting of a bundle of packets for convenience. Our goal is to find the optimal coding assignment that minimizes the total calculation cost while satisfying the required security level. In fact, minimizing calculation cost and maximizing security level are critical issues in nowadays networking systems which are ususally conflicting. The present proposed model aims to consider those conflicting issues simultaneously and proposes an evolutionary multiobjective optimization approach by assigning different weights to those two different objectives.

*1) Input parameters:* Table I summarizes the parameters and their meanings. Let $I$ denote the set of intermadiate nodes. F represents the set of packet items, m denotes the generation size which is equal to the number of packets per file, and n represents the packet size. In this paper, we perform coding accross one generation. $\delta_v$ is the number of vectors received

[2]Network coding with homomorphic signatures in NDN.

[3]With random linear network coding, data is represented as vectors.

[4]The solution could be easily generalized to the multi-sender, multi-receiver case if we use the signature scheme proposed in [3].

TABLE I
INPUT PARAMETERS AND DECISION VARIABLES IN MIP

| Parameter | Meaninig |
|---|---|
| $I$ | Set of intermediate nodes |
| $F$ | Set of packets |
| $m$ | Generation size |
| $n$ | Packet size |
| $\delta_v$ | Number of vectors received at $v \in I$ |
| $\rho$ | Security risk |
| $\Phi^c_{k,q}$ | The encoding cost of $\{p_1, p_2, ..., p_k\} \in \mathbb{F}^n_q$ |
| $\Phi^s_{k,q}$ | The homomorphic signature cost of $\{p_1, p_2, ..., p_k\} \in \mathbb{F}^n_q$ |
| $\Phi^v_{k,q}$ | The homomorphic verification cost of $\{p_1, p_2, ..., p_k\} \in \mathbb{F}^n_q$ |
| $\Phi^d_{k,q}$ | The decoding cost of $\{p_1, p_2, ..., p_k\} \in \mathbb{F}^n_q$ |
| $\mu$ | The security margin required |
| $\omega_i$ | A weight selected by the network designer to reflect the relative importance of an objective function |
| Decision variable | |
| $T_{v,m,q}$ | Binary variable indicating whetheher to operate linear cominations of $\{p_1, p_2, ..., p_m\} \in \mathbb{F}^n_q$ at $v \in I$ |
| $\eta_v$ | Number of linear combinations performed by $v \in I$ |

at $v \in I$. $\rho$ is the security risk encountered if we do not perform coding and signature verification. We assume the security level $\mu$ is given. $\omega_i$ denotes a weight selected by the network designer to reflect the relative importance of an objective function. While $\omega_0$ is the calculation cost weight coefficient and $\omega_1$ is the security level weight coefficient. The encoding, homomorphic signature, homomorphic verification and decoding costs of k packets $\{p_1, p_2, ..., p_k\} \in \mathbb{F}^n_q$ are represented respectively by $\Phi^c_{k,q}$, $\Phi^s_{k,q}$, $\Phi^v_{k,q}$ and $\Phi^d_{k,q}$.

*2) MIP model:* We present the MIP model for secure network coding assignment in named data networking. We assume the nodes have homogeneous calculation capacities, and the packets transmitted in the network are linearly independant. Let $\Phi^r$ denote the calculation cost of the producer node which is given by :

$$\Phi^r = m * (\Phi^c_{m,q} + \Phi^s_{m,q})$$

and $\Phi^t$ denote the calculation cost of the consumer node which is given by :

$$\Phi^t = \Phi^v_{m,q} + \Phi^d_{m,q}$$

With this coding policy, we describe our multiobjective function to minimize the total processing cost of all secure coding operations (1), and to maximize the security level in the whole network (2). If we consider $\rho$ as the security risk if we do not perform coding operations, then the security level in this case is equal to $\frac{1}{\rho}$. In fact, the security risk presents the propagation rate of erroneous messages. Thus, the notion of security level used in this paper refers to that of pollution. The higher the security level, the lower the polluton spread. We use weightening coefficients to reflect the relative importance of an objective function. (3) ensures the margin security authorized in the entire network. (4) assures that the maximum number of vectors received at each node is less or equal to the generation size. If a given node perform coding and signature operations than the number of linear combinations it should operate must be at least equal to one and at most equal to m (5) et (6). The total number of linear combinations created in the network do not exceed the generation size m (7).

$$min(\Phi^r + \sum_{v \in I} T_{v,\delta_v,q} * (\Phi^v_{\delta_v,q} + \eta_v * (\Phi^c_{\delta_v,q} + \Phi^s_{\delta_v,q})) + \Phi^t) \quad (1)$$

$$max \sum_{v \in I} (\frac{T_{v,\delta_v,q}}{1-\rho} + \frac{1 - T_{v,\delta_v,q}}{\rho}) \ \ subject\ to \quad (2)$$

$$\sum_{v \in I} ((1-\rho) * (1 - T_{v,\delta_v,q}) + \rho * T_{v,\delta_v,q}) < \mu \quad (3)$$

$$\delta_v \leq m, \ \ \forall v \in I \quad (4)$$

$$\eta_v \geq T_{v,\delta_v,q}, \ \ \forall v \in I \quad (5)$$

$$\eta_v \leq T_{v,\delta_v,q} * m, \ \ \forall v \in I \quad (6)$$

$$\sum_{v \in I} \eta_v = m \quad (7)$$

Our MIP finds the optimal solution for the coding and verification assignment model. An adaptation of the proposed coding model for NDN is required. This is done through the proposal of an Interest and a Data forwarding strategies.

### *B. Interest forwarding strategy*

The consumer sends a request containing mainly:

- The data name;
- The number of encoded messages required to reconstruct the original message : $r$.

The consumer proceed ordinarily as in the original sketch of NDN networking without network coding. It sends several interest messages, each one corresponds to a particular chunk. The Interest packet carries a name that identifies the desired data. Content Names are hierarchically structured, e.g. $ndn/lastversion/document/file1/chunk0$ where "/" is the boundary between name components. An Interest packet may contain the name of the content being requested or a name prefix, e.g. $ndn/lastversion$ is a prefix of $ndn/lastversion/document/file1/c0$. Once the Interest reaches a node that has the requested content, a Data packet is sent back via the reverse path. The name of a coded packet contains the name list of all the chunks used to construct the combination, e.g. $ndn/lastversion/docum/file1/(c0, c3, c4)$ is the combination name of three chunks $c0, c3, c4$.

Upon receiving an Interest message, an intermediate node

checks whether it has the requested content in its cache (1-4). In this case InCS() function returns true, then it supplies all the independant linear combinations which match the data name and updates the field $r$. If it has not delivered all the necessary packets (5-10), it transfers the request to other routers. To transfer the query, it must select the interfaces from the FIB[5] table (Select_ FIB), and update the field $r$ before sending the query (Update(interest)). OutCS(dataname) function returns the number of independant linear combinations stored at the corresponding node. While Update_ PIT precedure update the PIT[6] table with a new Interest entry.

**Algorithm 1** Interest Forwarding Strategy

**Input:** Interest
1: **if** InCS(dataname) **then**
2: $k \leftarrow$ OutCS(dataname);
3: $r \leftarrow r - k$;
4: **end if**
5: **if** $r \neq 0$ **then**
6: Select_ FIB(dataname,Interfaces);
7: Update(Interest);
8: Send(Interest,Interfaces);
9: Update_ PIT(Interest);
10: **end if**

### C. Data Forwarding Strategy

With random linear network coding (RLNC), the source breaks up data into several vectors (packets) and creates the augmentation coefficients.

$$Data = \begin{pmatrix} -\vec{\omega_1}- \\ -\vec{\omega_2}- \\ \vdots \\ -\vec{\omega_m}- \end{pmatrix} = \left(\begin{array}{c|cccc} -\vec{\gamma_1}- & 1 & 0 & \dots & 0 \\ -\vec{\gamma_1}- & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ -\vec{\gamma_m}- & 0 & 0 & \dots & 1 \end{array}\right)$$

The data packet contains mainly :

- The data name
- A linear combination of several vectors

$$Data = \sum_{i=1}^{l} \alpha_{ij} * \vec{\gamma_i}$$

- The augmentation coefficients
- The homomorphic signature

The Data packet corresponding to an Interest one could be the requested chunk itself or a coded (combined) packet containing the desired chunk. The coding assignment results obtained from the previous section will be used in the data forwarding processing. Algorithm 2 decsribes the data forwarding strategy. When a router receives a data message, there are two possibilities : whether it represents a compelling node or not. If $(T = 0)$, the actual node should just transfer the data message to the requesting interfaces provided by Select_ PIT procedure. Then, it must update the PIT and the FIB tables. Before forwarding the packet, it should store a copy of the data if it didn't have one in its content store. Otherwise if $(T = 1)$, then the node invokes Task_ Coding procedure which will verify the homomorphic signatures of received packets, perform a predefined number of linear combinations (output of the MIP model) and append homomorphic signatures to the created messages.

**Algorithm 2** Data Forwarding Strategy

**Input:** Data
1: **if** $T = 0$ **then**
2: Select_ PIT(dataname, interfaces);
3: Forward(Data, interfaces);
4: Update_ PIT(dataname);
5: Update_ FIB(dataname);
6: **if** InCS(dataname)=false **then**
7: CheckIndependance(Data);
8: Store(Data);
9: **end if**
10: **else**
11: Task_ Coding();
12: **end if**

## III. Evaluation results

In what follows, we consider three different scenarios. Firstly, we consider the case of applications that can not withstand a large computational burden for step-by-step verification and coding. In this case, we will perform comparision analysis of our proposed model called **OPT** with a trivial case where all the nodes in the network must do coding and homomorphic operations. This case was proposed in [11] and we refer to it with OPT-C. Secondly, we consider the case of applications that have a good computational power and require a high level of security with a hop-by-hop verification and signature. In this case, comparision will be held with a trivial case where all the nodes avoid doing coding and homomorphic signature operations. This variant corresponds to the proposed scheme in [1] and we refer to it with OPT-S. Thirdly, we aim to find a trade-off between both calculation cost and security level by adjusting weightening coefficients.

### A. Scenario and parameters setting

We consider arbitrary topologies (nodes uniformly distributed within a square area) with various network sizes (20, 40, 60, 80, 100). The security risk is set identical for all nodes and we specify this value. The number of vectors received at each node is distributed randomly across the network. We solved the MIP model using CPlex[7]. The evaluation benchmarks are generated using HElib library for homomorphic encryption [12] which implements an optimized version of the BGV (Brakerski-Gentry-Vaikuntanathan) fully homomorphic

[5]Routing table including the next hop information for prefix names.

[6]Table responsible for keeping track of the currently unsatisfied interest packets.

[7]http://www-01.ibm.com/software/integration/optimization/cplexoptimizer.

TABLE II
COMPARISION OF OPTIMAL OVERHEAD PROCESSING COSTS

| | $q = 2^2$ | | $q = 2^4$ | | $q = 2^8$ | |
|---|---|---|---|---|---|---|
| $\|N\|$ | OPT | OPT-C | OPT | OPT-C | OPT | OPT-C |
| 20 | 69.29 | 75.19 | 68.30 | 77.87 | 68.53 | 78.10 |
| 40 | 101.64 | 121.32 | 101.03 | 119.71 | 123.29 | 144.60 |
| 60 | 137.23 | 166.70 | 135.33 | 163.60 | 167.04 | 199.22 |
| 80 | 173.50 | 212.95 | 169.80 | 207.28 | 212.30 | 255.70 |
| 100 | 210.50 | 260.12 | 203.83 | 250.63 | 255.38 | 309.41 |

TABLE III
COMPARISION OF OPTIMAL SECURITY LEVEL

| | $\rho = 0.6$ | | $\rho = 0.7$ | | $\rho = 0.8$ | |
|---|---|---|---|---|---|---|
| $\|N\|$ | OPT | OPT-S | OPT | OPT-S | OPT | OPT-S |
| 20 | 40 | 33.33 | 43.81 | 28.57 | 55 | 25 |
| 40 | 100 | 66.67 | 133.33 | 57.14 | 200 | 50 |
| 60 | 150 | 100 | 200 | 85.71 | 300 | 75 |
| 80 | 200 | 133.33 | 266.67 | 114.28 | 400 | 100 |
| 100 | 250 | 166.67 | 333.33 | 142.85 | 500 | 125 |

encryption scheme. For network coding calculation measurements, we used Kodo [13] which is a high-performance erasure coding library.

### B. Results

Our testbed experiments captured the following metrics :

1) **Key generation time** : A measure of how long it takes to generate the private and public keys. In this paper, we suppose that public and private keys are certified using a set of trust authorities [8].
2) **Encoding time** : A measure of how long it takes to create a linear combination of packets.
3) **Decoding time** : A measure of how long it takes to decode a set of packets and to create the original data.
4) **Encryption time** : A measure of how long it takes to encrypt a plaintext message.
5) **Evaluation time** : A measure of how long it takes to evaluate a circuit[9].
6) **Decryption time** : A measure of how long it takes to decrypt a ciphertext.

We run the testbed experiments on an Intel(R) core(TM) i3-2328M with 2,20 HZ CPU performance and 6 GB of RAM. All software was run on the 64-bit ubunto 16.10 Linux distribution.

*1) Performance evaluation of cumulative processing overhead:* Let's denote $f_1$ the objective function to minimize the overall calculation cost and $f_2$ the objective function to maximize the security level. The optimal cost is given by :

$$OPT\_Cost = \omega_0 * f_0 + \omega_1 * f_1$$

Let's consider the first case where $\omega_0 = 1$ and $\omega_1 = 0$. Table II compares the optimal cost achieved for six different network sizes (20, 40, 60, 80, 100) of arbitrary topologies. For each size, we note the average of the results obtained from 100 randomly generated instances. We fix the generation size to 8 ($m = 8$) and the security risk to 0.8 ($\rho = 0.8$). For every instance, we compute the optimal cost when packets belong to the $\mathbb{F}_{2^2}$ finite field, the $\mathbb{F}_{2^4}$ finite field or the $\mathbb{F}_{2^8}$ finite field[10]. The size of a field is denoted $q$.

[8]This is the same case of Modern web browsers which integrate natively a list of certificates from different Certification Authorities.

[9]The linear combination function operated to signatures.

[10]A finite field (or Galois Field) is a mathematical construct where special rules are defined for the arithmetic operations.

In this case, we assume that all the nodes in the network perform verification operations. Nonetheless if we consider that only compelling nodes operate verifications the cumulative processing overhead cost will be the same even if we increase the network size (26.078s for $\mathbb{F}_{2^2}$ , 29.913s for $\mathbb{F}_{2^4}$ and 32.545s for $\mathbb{F}_{2^8}$). Hence, the improvement is in the order of 10%. We can observe that as the network size increases, the gain gap in processing overhead of OPT over OPT-C also increases. Indeed, for a 20-nodes network the improvement is in the order of 6s while it goes to 50s for a 100-nodes network with a $\mathbb{F}_{2^2}$ finite field . The gain increases as we increase $q$ value. In fact, increasing the field size will increase the probability of successful decoding. However, it will also lead to increased computational complexity which results in slower applications.

*2) Performance evaluation of security level:* In this case we aim to maximize security level. Assume $\omega_0 = 0$ and $\omega_1 = 1$. Let's take $q = 2^4$, $\mu = 100$ and $m = 8$. Table III compares optimal security level encountered for different security risk values ($\rho = 0.6$, $\rho = 0.7$ and $\rho = 0.8$) considering five different network sizes (20, 40, 60, 80, 100). We observe that as we increase security risk $\rho$, the security level increases for our proposed solution OPT, though it decreases for the worst case model considered OPT-S. In fact, increasing the security risk will lead to increase the security level of compelling nodes. The improvement is in the order of 1% for $\rho = 0.6$ while it goes to 4% for $\rho = 0.8$.

*3) Performance trade-offs:* In order to study, the performance trade-offs between processing overhead cost and security level, a set of performance evaluation tests have been operated. In each case we vary the weightening coefficients of the two objective functions. We suppose that $\mu = 100$, $q = 2^4$ and the topology size is equal to 100. Fig.2 illustrates the gain gap between multiobjective optimal solution and optimal cumulative calculation cost and between multi-objective optimal solution and optimal security level. We notice that as we increase the overhead cost weight the gap decreases, it goes from 323s to 179s for cumulative processing overhead, and from 36 to 179 for security level. This happens in a symmetrical way for the security coefficient. In Fig.3, we show that our solution presents a good compromise between the two considered trivial cases. We observe that our solution exhibits nearly the same optimal values as OPT-C and OPT-S by increasing the calculation weightening coefficient. Fig.4 plots a set of 50 nodes, and illustrates the results of our optimization

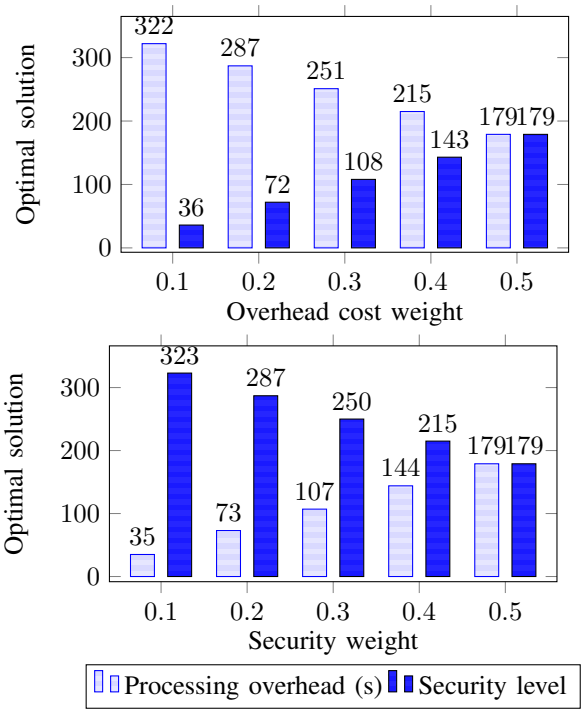


Fig. 1. Performance trade-offs.

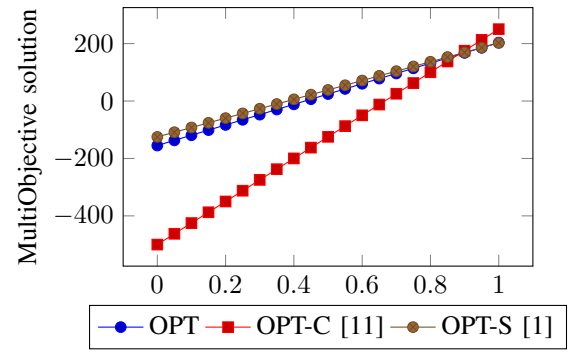


Fig. 2. Performance trade-offs of the multi-objective solution.

model. The x coordinate represents de node index, while the y coordinate represents the vectors number. In Fig.4(a), we consider that the set of compelling nodes should perform exactly 8 linear combinations. The model chooses the nodes holding the minimum number of received vectors. While in the Fig.4(b), we suppose that the number of linear combinations shared in the whole network must be equal to 16. In this case, our proposed model chooses 7 nodes.

## IV. Conclusion

Network Coding breaks with the forwording principle of conventional communication networks by allowing any network node to recombine several input packets into one coded packet. It represents an interesting technique which can provide throughput improvements and a high degree of robustness in packet networks. In this paper, we present a novel Mixed Integer Program for data coding and signature in named

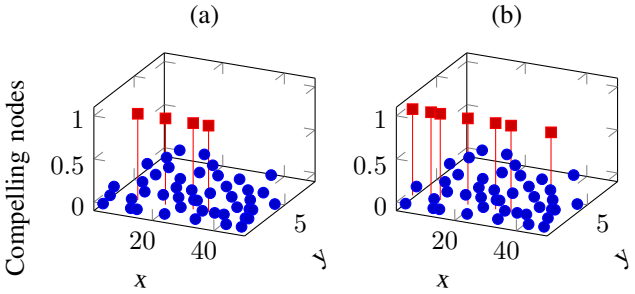


Fig. 3. Example of compelling nodes in a 50 nodes topology.

data networking. Our solution achieves better cost performance compared to existing solutions where all the nodes must perform coding, signature and verification operations. It reduces significantly the processing overhead generated by the homomorphic encryption scheme and achieves the best trade-off between communication overhead and computation overhead while ensuring packets authentication. The numerical experiments are insightful for the future design and implementation of a caching policy and a distributed algorithm for practical uses in named data networking.

## References


[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. Briggs, and R. Braynard, "Networking named content." *Commun. ACM*, vol. 55, no. 1, pp. 117–124, 2012.

[2] C. Fragouli and E. Soljanin, "Network coding fundamentals." *Foundations and Trends in Networking*, vol. 2, no. 1, 2007.

[3] S. Agrawal, D. Boneh, X. Boyen, and D. M. Freeman, "Preventing pollution attacks in multi-source network coding." *IACR Cryptology ePrint Archive*, vol. 2010, p. 183, 2010.

[4] D. Boneh, D. M. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in *Public Key Cryptography*, ser. Lecture Notes in Computer Science, vol. 5443. Springer, 2009, pp. 68–87.

[5] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin, "Secure network coding over the integers." *IACR Cryptology ePrint Archive*, vol. 2009, p. 569, 2009.

[6] G. Zhang and Z. Xu, "Combing ccn with network coding: An architectural perspective." *Computer Networks*, vol. 94, pp. 219–230, 2016.

[7] W. Liu, S.-Z. Yu, G. Tan, and J. Cai, "Information-centric networking with built-in network coding to achieve multisource transmission at network-layer." *Computer Networks*, vol. 115, pp. 110–128, 2017.

[8] Y. Liu and S.-Z. Yu, "Network coding-based multisource content delivery in content centric networking." *J. Network and Computer Applications*, vol. 64, pp. 167–175, 2016.

[9] J. W. Bos, K. E. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme." in *IMA Int. Conf.*, ser. Lecture Notes in Computer Science, M. Stam, Ed., vol. 8308. Springer, 2013, pp. 45–64.

[10] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks." in *WISEC*, D. A. Basin, S. Capkun, and W. Lee, Eds. ACM, 2009, pp. 111–122.

[11] S.-H. Lee, M. Gerla, H. Krawczyk, K.-W. Lee, and E. A. Quaglia, "Performance evaluation of secure network coding using homomorphic signature," in *IEEE International Symposium on Network Coding (NetCod)*, 2011, pp. 1–6.

[12] S. Halevi and V. Shoup, "Algorithms in helib." *IACR Cryptology ePrint Archive*, vol. 2014, p. 106, 2014.

[13] N. J. H. Marcano, M. V. Pedersen, P. Vingelmann, J. Heide, D. E. Lucani, and F. H. P. Fitzek, "Getting kodo: Network coding for the ns-3 simulator." in *WNS3*, T. Henderson, E. Gamess, B. Swenson, and H. Tazaki, Eds. ACM, 2016, pp. 101–107.