International Journal of Innovative
Computing, Information and Control
Volume **10**, Number **1**, February **2014**

 ISSN 1349-4198
pp. 317–327

A CVP-BASED LATTICE SIGNATURE SCHEME FOR NETWORK CODING

Tao Shang, Hengli Pei and Jianwei Liu

School of Electronic and Information Engineering
Beihang University
No. 37, Xueyuan Road, Haidian District, Beijing 100191, P. R. China
{ shangtao; liujianwei }@buaa.edu.cn; peihengli@126.com

Received December 2012; revised April 2013

Abstract. *Pollution attack is one of the main threats confronting network coding. The policy of detection and prevention of polluted messages is an important aspect of secure framework in a network based on network coding. Inspired by the idea of self/nonself discrimination in immune principle, signature is an effective approach to the discrimination of normal message and abnormal message. Most of the existent signature schemes cannot catch up with the rapid development of high-speed computers. To provide a high-security guarantee to network coding and lower the computing complexity induced by signature scheme, we introduce lattice theory to construct a secure signature scheme. Firstly, we propose a lattice-based signature scheme for network coding and its core algorithm SampleCVP which can ensure the randomness of output signature. Secondly, we stipulate its security to the hard problem CVP (Closest Vector Problem) on lattices. Security analyses show that the proposed scheme has a stronger unforgeability for the natural property of lattices than traditional signature schemes.*
Keywords: Network coding, Pollution attack, CVP, Lattice, Signature scheme

1. **Introduction.** Network coding [1] offers a new approach to data transmission instead of traditional multicast routing. Differing from traditional "store and forward" routing pattern, it allows a single intermediate node to combine blocks (divided from files generated by a source node) received and then send them to neighbor nodes. This novel approach can greatly enhance the throughput of multi-hop wireless networks such as wireless sensor networks for environment monitoring and wireless ad hoc networks for dynamic service. However, it is particularly vulnerable to pollution attack by which malicious nodes could inject invalid blocks to prevent the reconstruction of original files at destination nodes. Furthermore, due to the special way of data propagation based on network coding, a single invalid block could cause pollution diffusion to a whole network, which can be treated as a serious denial of service attack and destroy the survivability of the network.

Until very recently, some solutions have been proposed to defend against pollution attack in network coding. According to different theoretical foundations, these solutions can be classified into two types: information-theoretic solution and cryptographic solution [2]. Information-theoretic solution adds redundant information into network messages to ensure that destination nodes can correctly reconstruct data as long as the ratio of invalid messages to valid messages is sufficiently low. These techniques do not rely on any computational hypothesis, but impose some restrictions on the number of nodes that an adversary can destroy, the number of messages that can be modified, and the number of links that an adversary can eavesdrop. Moreover, the redundant information of messages causes extra cost, thus reducing the throughput of a network. Being much

more efficient and secure than information-theoretic solution, cryptographic solution is usually based on some computational assumptions, such as the assumption that attacker cannot work out some mathematical puzzles within finite time. In fact, they do not make any other restrictions on attackers, including the number of malicious nodes, the number of messages that can be modified, and the number of links that an adversary can eavesdrop. Such methods allow any node of a network to verify the received messages and drop them immediately once detecting attack. The cryptographic solutions have been paid considerable attention to ever since the first homomorphic signature scheme was proposed [3]. Almost all of these solutions are based on traditional cryptographic methods, and could be confronted with security menace with the rapid development of high-speed computers, especially in the emergence of quantum information and quantum computer.

The security problems found in network coding systems are quite similar to the ones encountered in Biological Immune Systems (BIS). BIS can successfully solve the problem of unknown virus detection. So Artificial Immune System (AIS) [4] is considered to defeat fast-proliferating computer viruses. Most of related works [5] simulated the concepts and mechanisms of BIS, and especially the IBM laboratory [6] used only signatures technique to simulate partial immune mechanisms and restrain virus spreading well. Obviously, signature is an effective approach to the discrimination of normal message and abnormal message in a network. It defines self and non-self by message signature and could be further used to find malicious nodes in a network. Thus a kind of new message signature and verification scheme is needed for network coding. It is lattice theory [7] that would provide a new approach to signature scheme for network coding.

Lattice-based cryptographic schemes are constructed on the basis of hard problems on lattices such as shortest basis problem (SBP), shortest vector problem (SVP), shortest independent vector problem (SIVP) and closest vector problem (CVP). They have the following advantages over traditional schemes based on number theories: 1) stronger security that can defend against the attacks launched by high-speed computers; 2) less computational complexity and higher computing speed; 3) fewer parameters with approximate secure strength. Early lattice-based cryptographic scheme is the GGH scheme proposed by Goldreich et al. [8], which is always the foundation of other lattice-based cryptographic schemes. This scheme is related directly to a certain hard problem on lattices, but lacks security proof. Nguyen and Regev [9] showed how to recover the entire secret key from a transcript of signatures of GGH. Gentry et al. [10] designed a new trapdoor function based on lattices and constructed the related signature scheme FDH (Fully Domain Hash) in which the signature satisfies Gaussian distribution, thus making it hard to obtain any information about secret key from the related signature. However, all of the schemes based on lattice theory cannot be used directly to prevent pollution attack in network coding. By improving the "key generation" process of FDH, Boneh and Freeman [11] constructed a lattice-based signature scheme with homomorphic property which can verify the linear combination of messages, but this scheme has a big restriction on the random coefficients and the number of messages is combined by a single node in network coding. To defend against pollution attack without making restrictions on parameters (such as random coefficients, the number of nodes), in this paper, we introduce lattice theory to construct a secure signature scheme for network coding.

The main contributions of our works are:

1) SampleCVP algorithm. We design an efficient algorithm that, when given a basis of an arbitrary lattice Λ and a random vector x whose dimension is same as Λ, it samples a lattice point from a discrete Gaussian distribution and the distance between the lattice point and vector x is less than a small value which will be discussed in Section 4.

2) CLS signature scheme. We propose a CVP-based lattice signature scheme (CLS). As it is built on basis of hard problem on lattices, it has stronger unforgeability when compared with traditional signature schemes based on number theories. Meanwhile, the distribution of the signature only depends on the length of vectors in the basis which represents the secret key in our schemes, thus making it much more secure than the previously proposed schemes (for example, the GGH scheme is insecure because its signatures leak information about the "shape" of the trapdoor basis).

This paper is structured as follows. In Section 2, we introduce the background on lattices. Section 3 mainly focuses on basic functions for signature scheme: TrapGen and SampleCVP. Then we describe the CLS signature scheme and give a detailed security analysis in Section 4. Section 5 is our conclusion.

2. Background on Lattices.

A lattice is defined informally as the set of all integral linear combinations of a set of vectors which are independent of each other [12]. It follows that such a lattice can be defined by an infinite number of bases. A formal definition can be seen in Definition 2.1.

Definition 2.1. [13] *A lattice Λ is a discrete sub-group of R^n, or equivalently the set of all integral combinations of n linearly independent vectors over R.*

$$\Lambda = z_1\bar{b}_1 + \cdots + z_n\bar{b}_n, \quad b_i \in R^n \tag{1}$$

where $z_1, \cdots, z_n \in Z$, $\bar{B} = (\bar{b}_1, \ldots, \bar{b}_n)$ is called the basis of Λ, and n is the dimension of Λ. $\Lambda_{\bar{B}}$ can be called as a lattice of the basis $\bar{B}$. Here integral lattice is considered and $(\bar{b}_1, \ldots, \bar{b}_n)$ are the vectors of integer elements.

The rank of a lattice is defined as the number of linearly independent vectors in any basis for the lattice. A full-rank lattice is defined as a lattice in the basis of which the number of linearly independent vectors is identical to the dimension of any vectors (which is also known as lattice point) in this lattice [14]. A simple instance of two-dimensional full-rank lattice is shown in Figure 1.

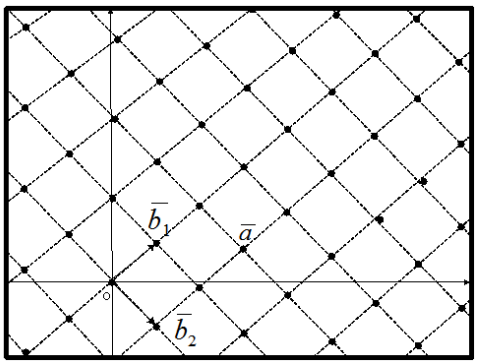


FIGURE 1. A simple instance of two-dimensional full-rank lattice ($\bar{b}_1$, $\bar{b}_2$ denote the vectors in a basis, $\bar{a}$ denotes a lattice point)

2.1. Hard random lattices.

Hard random lattices [15] are used frequently in lattice cryptography for their "worst-case hardness" [16] of hard problems on lattices. There are two kinds of lattice defined in the following:

Definition 2.2. *$\Lambda^{\perp}(A)$ is a kind of lattice when given a matrix A,*

$$\Lambda^{\perp}(A) = \{\bar{v} | A\bar{v} = 0, A \in Z^{n\times m}\} \tag{2}$$

Definition 2.3. *$\Lambda^{\bar{u}}(A)$ is a kind of lattice when given a matrix A and a vector $\bar{u}$,*

$$\Lambda^{\bar{u}}(A) = \{\bar{v} | A\bar{v} = \bar{u}, A \in Z^{n\times m}, \bar{u} \in Z^n\} \tag{3}$$

According to Definition 2.3, the lattice $\Lambda^{\bar{u}}(\mathrm{A})$ is constructed by using the vector $\bar{u}$, which implies that different vectors correspond to different lattices. Thus, the signatures of messages $\bar{E}_1, \ldots, \bar{E}_m$ transmitted in a network are the lattice points of different lattices $\Lambda^{\bar{E}_1}(\mathrm{A}), \ldots, \Lambda^{\bar{E}_m}(\mathrm{A})$. For this reason, if $\Lambda^{\bar{u}}(\mathrm{A})$ is used in our scheme, it will be very difficult to construct a standard process to verify these signatures of different lattices. Meanwhile, since $\Lambda^{\perp}(\mathrm{A})$ is not related to the messages transmitted in network, it can be easily adopted to build a signature scheme based on lattices.

2.2. **Hard problems on lattices.** There are four main hard problems on lattices [17]: shortest basis problem (SBP), shortest vector problem (SVP), shortest independent vector problem (SIVP) and closest vector problem (CVP). Micciancio and Goldwasser [18] presented a graphical representation of the relationship between these hard problems. It shows that both SBP and SIVP can be reduced to SVP, and SVP can be further reduced to CVP. Therefore, SVP and CVP are harder than the other two problems. It is more meaningful to explore the signature scheme based on harder problems on lattices for high security.

Definition 2.4. [19] *Given a lattice Λ, the shortest vector problem is to find a non-zero vector $\bar{v} \in \Lambda$, $\forall \bar{u} \in \Lambda$, $||\bar{v}|| \leq ||\bar{u}||$ ($||\bar{v}||$ refers to the Euclidean norm of the vector $\bar{v}$). Similarly, the approximately shortest vector problem is defined as given a lattice Λ, find a non-zero vector $\bar{v} \in \Lambda$, $\exists\gamma \in R^+$, $\forall \bar{u} \in \Lambda$, $||\bar{v}|| \leq \gamma||\bar{u}||$.*

Definition 2.5. [20] *Given a lattice Λ, let $\vec{t}$ be a vector in R^n, the closest vector problem is to find a vector $\bar{v} \in \Lambda$, $\forall \bar{u} \in \Lambda$, $||\bar{t} - \bar{v}|| \leq ||\bar{u} - \bar{v}||$. Similarly, the approximately closest vector problem is to find a non-zero vector $\bar{v} \in \Lambda$, $\exists\gamma \in R^+$, $\forall \bar{u} \in \Lambda$, $||\bar{t} - \bar{v}|| \leq \gamma||\bar{u} - \bar{v}||$.*

3. **Basic Functions.** In this section, we present two basic functions for signature scheme: TrapGen and SampleCVP. The TrapGen function can output a matrix A of uniform distribution and the basis T of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$, and the SampleCVP function samples a lattice point close to a given vector $\bar{x}$ from the lattice $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$.

Definition 3.1. [21] *Let $C > 0$ and $\delta > 0$ be constants and let $q \geq 3$ be odd. Moreover, let $m_1 \geq d = (1+\delta)n \lg q$, $m_2 \geq d = (4+2\delta)n \lg q$ and $m = m_1 + m_2$, $n \in Z^+$. TrapGen is a function in which exists a PPT (Probabilistic Polynomial Time) algorithm that outputs $(A, T) \in Z_q^{n\times m} \times Z^{m\times m}$ with special properties (T is a "good basis" of A).*

Definition 3.2. *Combining Babai's nearest plane algorithm* [22] *and SampleD algorithm* [10]*, we construct the SampleCVP algorithm which can sample from a discrete Gaussian distribution $D_{\Lambda,s,\bar{c}'}$ (Λ is the domain to be sampled from, s is the variance of the distribution while $\bar{c}'$ denotes the mean value). The input to SampleCVP is a "good basis" of a m-dimensional lattice Λ, a parameter $s \in R^+$ and a vector $\bar{x} \in R^m$. We describe the algorithm as if it has access to an oracle that can sample exactly from $D_{R,s,c}$ by means of inverse transform sampling method.*

In a figurative way, we describe these algorithms by a two-dimensional lattice from Figure 2 to Figure 4. The structure line of the lattice is ignored. Figure 2 shows the Babai's nearest plane algorithm. $\{\bar{t}_1, \bar{t}_2\}$ denotes the basis of the lattice and $\bar{x}$ is the input vector. The objective is to find a vector close to $\bar{x}$ in the lattice spanned by $\{\bar{t}_1, \bar{t}_2\}$. This algorithm includes three steps: firstly, it projects $\bar{x}$ on $\bar{t}_1$ to obtain $r_1\bar{t}_1$ and then rounds off r_1 to z_1 to obtain the vector $z_1\bar{t}_1$; secondly, let $\bar{p}_1$ be $\bar{x} - z_1\bar{t}_1$, then project $\bar{p}_1$

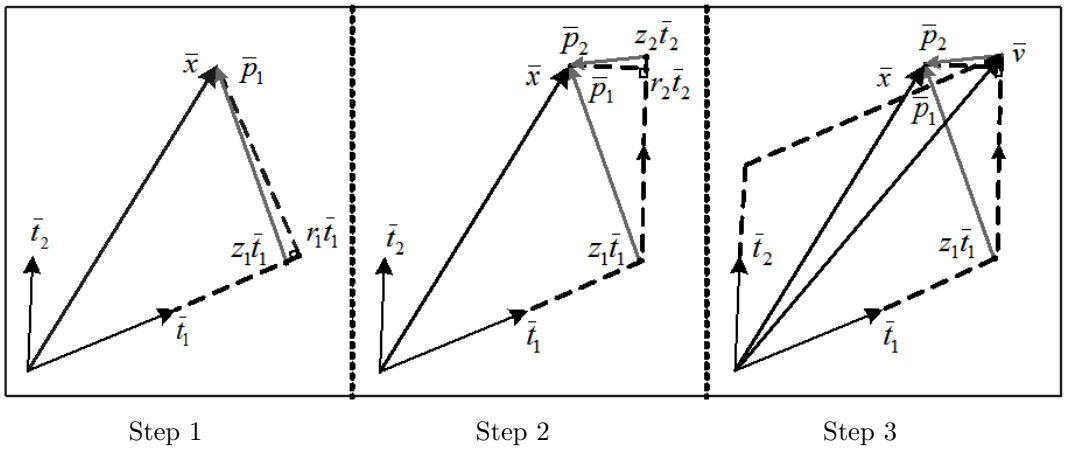


FIGURE 2. Babai's nearest plane algorithm

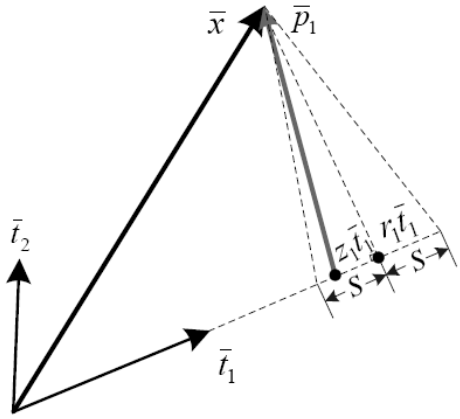


FIGURE 3. The difference between Babai's nearest plane algorithm and SampleD

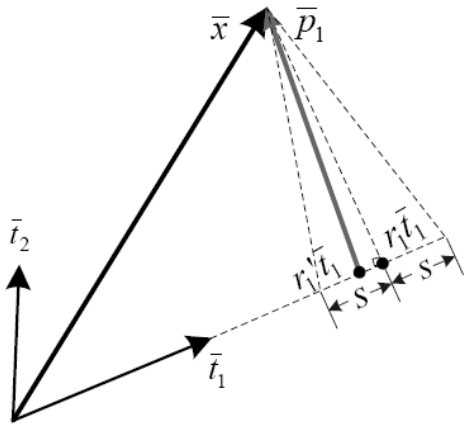


FIGURE 4. The difference between SampleCVP and SampleD

on $\bar{t}_2$ to obtain $r_2\bar{t}_2$, rounds off r_2 to z_2 to obtain the vector $z_2\bar{t}_2$, and thus obtain the vector $\bar{p}_2$; thirdly, return the output vector $\bar{v}$ by using $\bar{x} - \bar{p}_2$. Obviously, the distance between $\bar{x}$ and $\bar{v}$ approaches 0 if $\bar{t}_1$ is orthogonal to $\bar{t}_2$. It is a very important incentive in lattice-based cryptographic scheme that quite a few existing schemes use a "good basis" (sufficiently orthogonal) as the secret key to produce the signature $\bar{v}$ (e.g., the GGH scheme). Security is based on the knowledge that even a LLL-reduced basis of a random

basis can only achieve a vector which is $2^{n/2}l$ (l is the least distance between a lattice point and $\bar{x}$) far from $\bar{x}$, thus making it hard for an attacker to break this scheme. However, as the output $\bar{v}$ is a definite value when given a fixed basis, it might leak information about the "shape" of the basis and the original messages. That is why GGH was broken by Craig Gentry in [23].

In case of being broken through leaking information about secret key and original messages, Gentry et al. [10] proposed a new sampling algorithm called SampleD whose output is sampled from a Gaussian distribution. The different step compared with Babai's nearest plane algorithm is shown in Figure 3. Instead of rounding off r_1 to z_1 directly, it constructs a core algorithm SampleZ (the only difference between Babai's algorithm and SampleD) to sample z_1 randomly from all the integer values surrounding r_1.

The drawbacks of SampleD are apparent due to its SampleZ algorithm:

1) Low efficiency. As SampleZ is a probabilistic algorithm that outputs value with the overwhelming probability, it could not obtain a return value in the case of improper parameters.

2) Low security. As the value of the variance s in SampleD should be higher than the multiplication of a small value and the maximum length of all the Gram-Schmidt vectors $\bar{t}_i$ of the basis T, the distance between the resulting vector $\bar{v}$ and $\bar{x}$ might be lengthened due to this restriction, which would make it easier for an attacker to find a vector close to $\bar{x}$ without using SampleD. As the sampling algorithm is the core part of signature scheme, it might become more convenient for an attacker to forge a signature by means of the SampleD algorithm.

Considering these drawbacks induced by SampleZ, we construct a more efficient sampling algorithm called SampleCVP which does not impose any restriction on s and can return a value with the probability of 100%.

The difference between SampleCVP and SampleD is shown in Figure 4. Instead of choosing z_1 directly, the improved SampleZ firstly samples a real value r_1' surrounding r_1 from a Gaussian distribution by means of inverse transform sampling method and then rounds off it to an integer z_1.

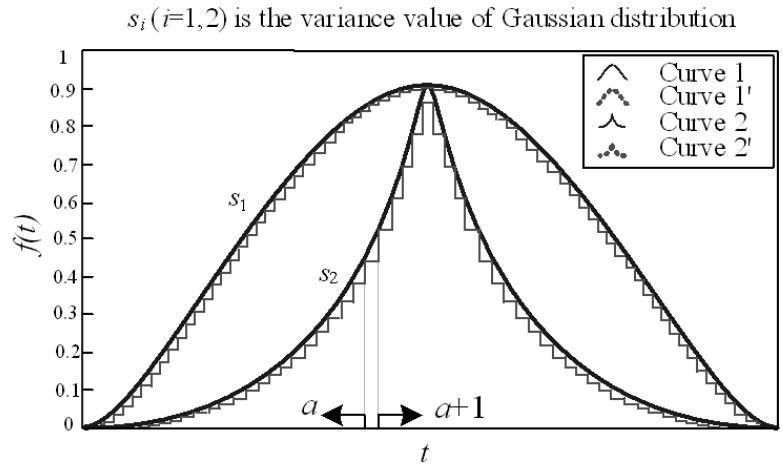
According to the SampleCVP algorithm, the output vector $\bar{v}$ can be written as $\bar{v} = z_1\bar{t}_1 + \cdots + z_n\bar{t}_n$. As we have described in former sections, the objective of SampleCVP is to find a vector which is sampled from a Gaussian distribution, and what is more, z_i is rounded off by r_i, so that its distribution becomes obscure.

Theorem 3.1. *The distribution of the output vector $\bar{v}$ of SampleCVP algorithm approaches Gaussian distribution.*

Proof: It is obvious that $z_1\bar{t}_1, \cdots, z_n\bar{t}_n$ is independent of each other for the property of the basis in a lattice. Therefore, to prove the distribution of $\bar{v}$ suffices Gaussian, we have to demonstrate that distributions of the n variables $z_1, \cdots, z_n$ suffice Gaussian. It is obvious that the value which is rounded off from a variable sufficing Gaussian distribution also approaches Gaussian distribution. Consider a one-dimensional Gaussian distribution shown in Figure 5, the solid line (Curve 1) denotes the Gaussian probability density function of r_i'. All the values between an integer a and $a+1$ are rounded off to be a, so the Gaussian probability density of z_i should be the average value between a and $a+1$ on Curve 1. Let the probability density be denoted by $valueZ$, then it can be calculated by:

$$valueZ \times 1 = \int_a^{a+1} f(t)dt \tag{4}$$

Then we obtain the step curve in Figure 5. Obviously, the larger the value of s is, the more precisely the step curve approaches Gaussian distribution. Also, the smaller the



Curve 1 and Curve 2 denote the distribution of r_i' with different variance values s_1 and s_2 ($s_1 > s_2$). Curve 1′ and Curve 2′ denote the distribution of z_i which is rounded off from r_i'.

FIGURE 5. One-dimensional Gaussian distribution

distance between $\bar{x}$ and $\bar{v}$ is, the more secure our signature scheme is. Meanwhile, the value of s is proportional to the distance between $\bar{x}$ and $\bar{v}$. If the value of s is larger, the security of signature scheme will become weak. Thus there exists a "trade-off" between the distance between $\bar{x}$ and $\bar{v}$ and the precise distribution of $\bar{v}$. Here, we will not discuss the optimal value for s, but just give out the originality of this idea.

4. CVP-Based Lattice Signature Scheme for Network Coding.

4.1. Secure framework of network coding.

We present a high-level description of random linear network coding for the reason of its most wide use among all of network coding schemes. Figure 6 shows the common topology of network coding, a single-source multicast network. The file originated by a source node is divided into blocks to be transmitted in the network. To defend against pollution attack which can interrupt reconstruction of those files, each block should be attached with its corresponding signature.

In this settings, S denotes the source node which generates original files to be reconstructed, and a subset of nodes described as $t_1 \cdots t_k$ denote destination nodes. The function of signature is to guarantee that the files generated by S can be reconstructed correctly by the destination nodes $t_1 \cdots t_k$ without any attacks launched by malicious nodes.

In our secure framework of network coding, S firstly creates m augmented vectors (also called messages) defined as

$$\bar{E}_i = (\bar{u}_i \,\|\bar{x}_i) = (\underbrace{0,\ldots,0}_{i-1},1,\underbrace{0,\ldots,0}_{m-i}\|\bar{x}_i), \quad i = 1,\ldots,m \tag{5}$$

where the m original vectors $\bar{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ constitute the original file F which will be reconstructed by destination nodes. Each $\bar{x}_i$ is pre-pended with the vector $\bar{u}_i$ of m dimensions containing a single "1" in the ith position to formulate the augmented vectors $\bar{E}_i$. To defend against pollution attack, S appends signatures to each augmented vectors by using signature generation algorithm and then sends the blocks denoted as $\{\bar{E}_i \,\|\bar{S}_i\}$ in this context to its neighbor nodes.

Each intermediate node (denoted as I) in the network firstly checks the validity of the signature $\bar{S}_i$ in the received block $\{\bar{E}_i \,\|\bar{S}_i\}$ and then puts the augmented vectors that

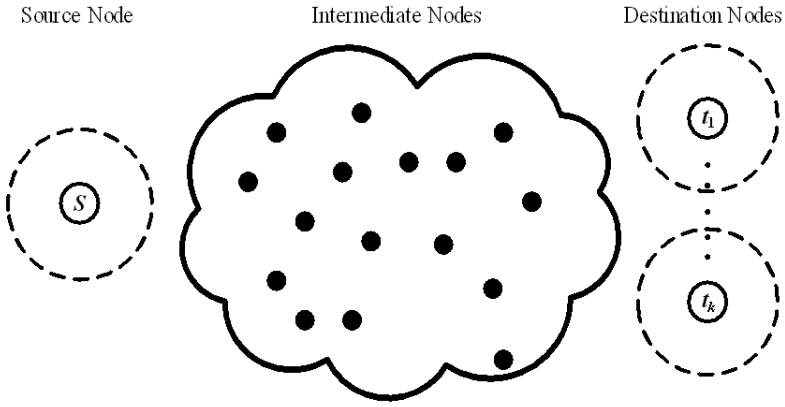


FIGURE 6. Network topology

have passed verification into its local storage. Once there are k vectors in its local storage, I computes a new vector $\bar{w}$ through a linear combination of the k messages in its local storage, and then produce the signature for the combined message. We say a vector $\bar{w}$ is valid if it lies in the linear span of the original augmented vectors $\bar{E}_1, \ldots, \bar{E}_m$. It is evident that if all nodes follow the protocol honestly, then every messages transmitted in the network are valid. In most cases, in order to avoid the transmission of plaintexts in a network, source node has to produce m combinations of the m original messages, which is the same as what intermediate nodes do.

Similarly, destination nodes process received blocks identically with what intermediate nodes do. They firstly check the validity of received signatures and then put the augmented vectors that have been verified into its local storage. To recover the original file F, a destination node must store m blocks $\{\bar{E}_i \left\| \bar{S}_i = \bar{u}_i \right\| \bar{x}_i \left\| \bar{S}_i \mid i = 1, \cdots, m\right\}$ in which $\bar{E}_1, \ldots, \bar{E}_m$ are linearly independent. We define U as a matrix whose rows are vectors $\bar{u}_1, \ldots, \bar{u}_m$ and define another matrix X whose rows are vectors $\bar{x}_1, \ldots, \bar{x}_m$, then the original file F can be reconstructed by $F = U^{-1}X$.

Introducing signatures to network coding can ensure the verification of received blocks at intermediate nodes, thus we can point out invalid blocks immediately without transmitting it to destination nodes and consequently reduce the cost on transmission of invalid messages.

4.2. **Signature scheme.** Based on SampleCVP and secure framework of network coding, we construct the CVP-based Lattice Signature scheme (CLS) which includes three core parts: KeyGen, Sign and Verify. The input of CLS includes four parameters: $C > 0$, $\delta > 0$, an odd $q \geq 3$ and the original message $\bar{x}$. Details are described as follows.

KeyGen: use TrapGen(C, δ, q, n) to generate a public key A and a private key T. Here T is a "good basis" of $\Lambda^{\perp}(A)$.

Sign: choose an arbitrary hash function H that can project a vector from n dimensions to m dimensions. Calculate $H(\bar{x})$. Then, find a vector $\bar{v}$ close to $H(\bar{x})$ in lattice $\Lambda^{\perp}(A)$ by means of SampleCVP.

Verify: upon receiving $\bar{x}$ and the related signature $\bar{v}$, calculate $H(\bar{x})$ and decide whether $\|H(\bar{x}) - \bar{v}\| \leq \rho$ (ρ is a value much smaller than $2^{m/2}l$ which is defined in Babai's nearest plane algorithm [22]). If satisfied, calculate $A\bar{v}$. If the result equals 0, accept the message, otherwise, drop it and wait for the arriving of next message.

4.3. **Parameter selection.** In cryptography, the security parameter is usually a variable that measures the input size of a problem. Both the parameters of the cryptographic algorithm or protocol as well as the adversary's probability of breaking security are expressed in terms of the security parameter.

In our signature scheme, the security parameters are the dimension n of the input message $\bar{x}$ and the variance s which is used in SampleCVP. The other parameters include C, δ, q, m, ρ and the parameters of network coding. C, δ, q, m in the TrapGen function should be 5, 0, n^3, $5n\lg q$ respectively [7] to ensure the correctness of the distribution of A and T. The distance ρ between message $\bar{x}$ and its signature $\bar{v}$ should be $\sqrt{5n\lg q}([s]+1/2)MAX_{i=1,\ldots,m}(l_{i1})$ to ensure the security of CLS. It is decided according to Theorem 4.1.

Theorem 4.1. *The value of ρ is $\sqrt{5n\lg q}([s]+1/2)MAX_{i=1,\ldots,m}(l_{i1})$ so as to ensure the security and correctness of CLS.*

Proof: According to the procedure of CLS, the signature of an arbitrary message $\bar{x}$ is $\bar{v}=T[T^{-1}\times H(\bar{x})\pm s]$, so the distance ρ between $H(\bar{x})$ and its signature $\bar{v}$ is

$$\rho=||H(\bar{x})-\bar{v}||_2=||T[T^{-1}H(\bar{x})\pm s]-H(\bar{x})||_2 \tag{6}$$

Then we can calculate the value of ρ as follows:

$$\begin{aligned}\rho&=||T[T^{-1}H(\bar{x})\pm s]-H(\bar{x})||_2=||T\{[T^{-1}H(\bar{x})\pm sI]-T^{-1}H(\bar{x})\}||_2\\&\le||T\{T^{-1}H(\bar{x})+1/2+[sI]-T^{-1}H(\bar{x})\}||_2=||T([sI]+1/2)||_2\\&\le([s]+1/2)||\begin{pmatrix}t_{11}+\cdots+t_{1m}\\ \cdots\\ t_{m1}+\cdots+t_{mm}\end{pmatrix}||_2=([s]+1/2)||\begin{pmatrix}l_{11}\\ \cdots\\ l_{1m}\end{pmatrix}||_2\\&\le\sqrt{m}([s]+1/2)MAX_i(l_{i1})=\sqrt{5n\lg q}([s]+1/2)MAX_{i=1,\ldots,m}(l_{i1})\end{aligned} \tag{7}$$

As $\|H(\bar{x})-\bar{v}\|\le\rho$ is necessary in the verify process of CLS, to ensure the completeness [7] of the scheme, ρ should be at least $\sqrt{5n\lg q}([s]+1/2)MAX_{i=1,\ldots,m}(l_{i1})$. Consider that the larger value of ρ will result in the less security of our scheme, it is optimal to set ρ to be $\sqrt{5n\lg q}([s]+1/2)MAX_{i=1,\ldots,m}(l_{i1})$.

Although there are also the other parameters of network coding (e.g., the number of nodes and the random coefficient of network coding) used in the process which is shown in Section 4.1, they are not related to the performance of CLS. Consequently, there exist two important security parameters n and s in our signature scheme.

4.4. **Security analysis.** With the restrictions on some parameters, the security of CLS can be described as follows:

Theorem 4.2. *Let n and s be the security parameters. CLS is (t,q_{sig},ε) unforgeable if CVP $(5n\lg q, 2\sqrt{5n\lg q}([s]+1/2)MAX_{i=1,\ldots,m}(l_{i1}))$ is (t',ε')-hard with $t'=t+(q_{sig}+1)(T_{A\bar{x}}+T_h)$, $\varepsilon'=\varepsilon-2^{-\omega\lg(5n\lg q)}$. ($CVP(a,b)$ denotes the hard problem CVP in which a denotes the dimension of the input message and b denotes the distance between the the input message and the output signature, (t',ε') means that the adversary can use at most t' to solve the hard problem with the probability of at least ε').*

Proof: We assume that there exists a successful forger $\mathcal{A}$ against CLS who can forge a valid signature for any polluted message $\bar{x}^*$, and we construct an algorithm $\mathcal{B}$ via a black box simulation, such that $\mathcal{B}$ solves the CVP$(5n\lg q, 2\sqrt{5n\lg q}([s]+1/2)MAX_{i=1,\ldots,m}(l_{i1}))$ with the probability ε' in the time t'. In the process, $\mathcal{A}$ uses Hash queries and signature queries to forge a valid signature for an arbitrary message, the two queries are defined as follows.

Hash oracle queries. On inputting a message $\bar{x}$, $\mathcal{B}$ looks up $\bar{x}$ in its local storage. If it finds a tuple $(\bar{x}, h, \bar{v}_{\bar{x}})$, then returns h. Otherwise, $\mathcal{B}$ randomly chooses a value $\bar{v}_{\bar{x}}$ among all the solutions to $A\bar{x} = 0$ as the signature of $\bar{x}$ (The time cost denoted as $T_{A\bar{x}}(n)$), and then finds a hash value h from all the hs sufficing $||h - \bar{v}_{\bar{x}}||_2 \leq \rho$ (The time cost denoted as $T_h(n)$). Then, stores $(\bar{x}, h, \bar{v}_{\bar{x}})$ and returns h.

Signature queries. On inputting a message $\bar{x}$, $\mathcal{B}$ uses Hash oracle queries to obtain the signature $\bar{v}_{\bar{x}}$, and returns $(\bar{x}, \bar{v}_{\bar{x}})$.

With these two kinds of query, $\mathcal{A}$ firstly makes q_{sig} queries on several messages it chooses and obtains tuples in form of $(\bar{x}, \bar{v}_{\bar{x}})$. After several times' queries, $\mathcal{A}$ stops and returns a valid forgery $(\bar{x}^*, \bar{v}^*)$ with $A\bar{v}^* = 0$ and $||H(\bar{x}) - \bar{v}^*||_2 \leq \rho$. Although $\mathcal{A}$ cannot query directly on $\bar{x}^*$, but it can query on the hash value of $\bar{x}^*$, so there is a tuple $(\bar{x}^*, h, \bar{v}_{\bar{x}^*})$ in the storage with $h = H(\bar{x}^*)$. As $||H(\bar{x}) - \bar{v}^*||_2 \leq \rho$, $||H(\bar{x}) - \bar{v}_{\bar{x}^*}||_2 \leq \rho$ and $A\bar{v}^* = A\bar{v}_{\bar{x}^*} = 0$, obviously, we can get $||\bar{v}_{\bar{x}^*} - \bar{v}^*||_2 \leq 2\rho$. Thus, we solve the $\mathrm{CVP}(5n \lg q, 2\sqrt{5n \lg q}([s] + 1/2)MAX_{i=1,\ldots,m}(l_{i1}))$.

However, we should notice that if $\bar{v}^* = \bar{v}_{\bar{x}^*}$, the process fails. In the following, we show that $\bar{v}^* = \bar{v}_{\bar{x}^*}$ holds but with negligible probability: by the minimum conditional entropy $\omega(\lg n)$ of $\bar{v}^*$ [7], we infer that $\bar{v}^* = \bar{v}_{\bar{x}^*}$ with probability at most $2^{-\omega \lg(5n \lg q)}$, which is negligible. Thus, we get that $\varepsilon' = \varepsilon - 2^{-\omega \lg(5n \lg q)}$. According to the procedure, it is obvious that $t' = t + (q_{sig} + 1)(T_{A\bar{x}}(n) + T_h(n))$.

5. **Conclusions.** The immune principle provides an important guide for the security mechanism of network coding. To enhance the adaptation and diversity of signature, based on lattice theory, this paper constructed a secure signature scheme for network coding to defend against pollution attack launched by malicious nodes. The signature scheme CLS was built on a newly sampling algorithm SampleCVP which can ensure the randomness of its output signature. To implement the signature scheme, we constructed the basic functions for signature scheme CLS, discussed the sampling algorithm of lattice point in detail, and proved that the security of this scheme can be reduced to CVP on lattices with security parameters. Furthermore, we can discuss the optimization of "good basis" for SampleCVP, and also can use the lattice signature to identify the malicious nodes for stronger immune function.

Acknowledgment. We would like to thank the National Key Basic Research Program of China (No. 2012CB315905), the National Natural Science Foundation of China (No. 61272501), and the Beijing Natural Science Foundation (No. 4132056) for valuable helps.

REFERENCES

[1] R. Ahlswede and N. Cai, Network information flow, *IEEE Transactions on Information Flow*, vol.46, no.4, pp.1204-1216, 2000.

[2] Z. Cao and Y. Tang, Survey on secure network coding, *Journal of Computer Applications*, vol.2, no.1, pp.499-505, 2010.

[3] R. Johnson, D. Molnar, D. Song and D. Wagner, Homomorphic signature schemes, *Lecture Notes in Computer Science*, vol.2271, no.1, pp.204-245, 2002.

[4] L. Castro and J. I. Timmis, Artificial immune systems as a novel soft computing paradigm, *Soft Computing Journal*, vol.7, no.8, pp.526-544, 2003.

[5] T. Li, X. Liu and H. Li, An immune-based model for computer virus detection, *CANS 2005*, vol.3810, no.1, pp.59-71, 2005.

[6] J. Kephart and W. Arnold, Automatic extraction of computer virus signatures, *Proc. of the 4th International Virus Bulletin Conference*, vol.1994, no.1, pp.178-184, 1994.

[7] G. Birkhoff, Lattices theory, *Coll. Pub.*, vol.1967, no.25, pp.3-15, 1967.

[8] O. Goldreich, S. Goldwasser and S. Halevi, Public-key cryptosystems from lattice reduction problems, *Lecture Notes in Computer Science*, vol.1294, no.1, pp.112-131, 1997.
[9] P. Q. Nguyen and O. Regev, Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures, *Journal of Cryptography*, vol.22, no.2, pp.139-160, 2009.
[10] C. Gentry, C. Peikert and V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, *Proc. of the 40th Annual ACM Symposium on Theory of Computing*, vol.2008, no.1, pp.197-206, 2008.
[11] D. Boneh and D. M. Freeman, Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures, *Lecture Notes in Computer Science*, vol.6571, no.1, pp.1-16, 2011.
[12] O. Regev, Lattice-based cryptography, *Lecture Notes in Computer Science*, vol.4117, no.1, pp.131-141, 2006.
[13] Z. Tian and S. Qiao, A complexity analysis of a Jacobi method for lattice basis reduction, *Proc. of the 5th International Conference on Computer Science and Software Engineering*, vol.2012, no.1, pp.53-60, 2012.
[14] D. Micciancio, The geometry of lattice cryptography, *Lecture Notes in Computer Science*, vol.6858, no.1, pp.185-210, 2011.
[15] P. Thomas, S. Willy, W. K. Than and H. Qiong, Efficient lattice-based signature scheme, *International Journal of Applied Cryptography*, vol.1, no.2, pp.120-132, 2008.
[16] M. Rose, *Lattice-Based Cryptography: A Practical Implementation*, Master Thesis, 2011.
[17] C. Gentry, Fully homomorphic encryption using ideal lattices, *Proc. of the 41st Annual ACM Symposium on Theory of Computing*, vol.2009, no.1, pp.169-178, 2009.
[18] D. Micciancio and S. Goldwasser, Complexity of lattice problems: A cryptographic perspective, *Kluwer Academic Publishers*, vol.2002, no.1, pp.3-15, 2002.
[19] O. Goldreich, On the limits of nonapproximability of lattice problems, *Journal of Computer and System Science*, vol.60, no.1, pp.540-563, 2000.
[20] D. Micciancio, The hardness of the closest vector problem with preprocessing, *IEEE Transactions on Information Theory*, vol.47, no.3, pp.1212-1215, 2001.
[21] J. Zhang, Hierarchical identity-based broadcast encryption scheme on lattices, *Proc. of 2011 the 7th International Conference on Computational Intelligence Security*, vol.212, no.1, pp.944-948, 2011.
[22] O. Regev, Lecture on CVP algorithm, *Lattices in Computer Science*, Tel Aviv University, 2004.
[23] C. Gentry, J. Jonsson, J. Stern and M. Szydlo, Cryptanalysis of the NTRU signature scheme (NSS), *Proc. of Asiacrypt*, vol.2248, no.1, pp.1-20, 2001.