International Scholarly Research Network
ISRN Communications and Networking
Volume 2011, Article ID 898254, 10 pages
doi:10.5402/2011/898254

Research Article

Group-Based Reduction Schemes for Streaming Applications

Riccardo Bernardini, Roberto Cesco Fabbro, and Roberto Rinaldo

DIEGM, University of Udine, Via delle Scienze 208, 33100 Udine, Italy

Correspondence should be addressed to Riccardo Bernardini, riccardo.bernardini@uniud.it

Received 4 May 2011; Accepted 3 July 2011

Academic Editors: A. K. Agrawala, G. Hasegawa, E.-C. Park, and A. B. Sesay

Some peer-to-peer streaming systems make use of linear codes to reduce the rate of the data uploaded by peers with limited upload capabilities. Such "data reduction" techniques are based on a vector-space approach and produce the data to be uploaded by means of linear combinations of the content data in a suitable finite field. In this paper, we propose a more general approach based on group theory. The new approach, while including the vector space approach as a special case, allows to design schemes that cannot be modeled as linear codes. We analyze the properties of the schemes based on the new approach, showing also how a group-based scheme can be used to prevent stream poisoning and how a group-based scheme can be converted into a secret-sharing scheme. Examples of group-based schemes that cannot be described in the vector-space framework are also shown.

1. Introduction

A problem that is currently attracting attention in the research community is the problem of streaming live content to a large number of nodes. The main issue to be solved is due to the amount of upload bandwidth required to the server that, unless multicast is used, is equal to the bandwidth required by a single viewer multiplied by the number of viewers. Although multicast is a possible solution, it has its drawbacks too, especially if the audience is spread among several different autonomous systems (AS).

An approach that recently attracted interest in the research community is the use of peer-to-peer (P2P) solutions. With the P2P approach, each viewer resends the received data to other users, and, ideally, if each user retransmitted the video to another user, the server would just need to "feed" a handful of nodes, and the network would take care of itself. Unfortunately, the application of the P2P paradigm to multimedia streaming has some difficulties. Maybe the most important one is due to the fact that the typical residential user have enough *download* bandwidth to receive the stream but not enough upload bandwidth to retransmit it. This makes the application of the P2P paradigm to video streaming not trivial.

Some peer-to-peer streaming systems [1–5] propose the use of linear codes (someone interprets this approach as an instance of network coding [6]) to overcome the asymmetric bandwidth problem. In order to adapt the upload bandwidth to the user capabilities, the node combines the content data by means of some linear combinations and forwards the result. If the node has a reduced upload bandwidth, the forwarded linear combinations will not be sufficient to recover the original content, but a node can contact more than one peer to receive different sets of linear combinations in order to be able to recover the content data.

The approaches proposed in [1–5] reduce the required data rate by using linear codes obtained as linear transformations of vector spaces over a finite field. The goal of this paper is to introduce a more general approach on data rate reduction based on group theory. We will show that the classical vector space approach is just a specialization of the theory presented here, since vector spaces are just a special type of groups. However, since groups are more general than vector spaces, the theory presented in this paper allows one to create new coding procedures that cannot be described as linear combinations in suitable vector spaces. Note that the only hypothesis required is that the groups involved have a finite number of elements, and in particular, it is not required that the groups are commutative.

Although the application that motivated this work was rate reduction for P2P streaming, we will show that the

theory presented here has a wider application range and it allows, for example, the construction of systems that counteract poisoning attacks [7] or allow secret sharing [8, 9].

This paper is organized as follows: in Section 2, we introduce a formalism for group-based reduction schemes (GBRS), in Section 3, we study the properties of GBRS; in Section 4, we give some examples of GBRS that cannot be described with the vector space approach; in Section 5, we give the conclusions.

2. Group-Based Reduction Schemes

Some P2P systems for multimedia streaming solve the problem of the limited bandwidth of residential users by uploading, instead of the whole content, some linear combinations (in a suitable finite field) of the data that constitute the content [1–3, 5]. The goal of this section is to introduce an alternative description of this type of "data reduction" procedures not based, as usual, on vector spaces but on group theory. In order to make the introduction of the group theory approach easier, we first introduce (in Section 2.1) a very general formalization of the reduction process that will be specialized in Section 2.2 to the desired group-based formalization.

For the sake of concreteness, we will refer to the peer-to-peer streaming application, but this does not prevent the application of the presented theory to other applicative contexts of network coding.

2.1. Data Reduction Schemes. We will model the *content stream* to be transmitted as a sequence of *content symbols* belonging to a finite *content alphabet* G. Each content symbol $g \in G$ requires, clearly, $\log_2|G|$ per element. The idea for reducing the rate necessary for the uploaded stream is to map every content symbol g to a *reduced symbol* u belonging to a smaller alphabet. Since the alphabet of u is smaller than G, the number of bits required for u will be smaller.

Definition 1. A reduction scheme is given via (i) a finite set S and (ii) a set of reduction functions $r_s : G \to K_s$, indexed by $s \in S$ and sharing the same domain G (the content alphabet).

Remark 1. Note that no constraint is put on K_s. In a practical context, it is expected that each K_s has smaller cardinality than G.

A node with limited upload bandwidth chooses at startup a *reduction parameter* $s \in S$. Every time it receives a new content symbol g, it reduces it by processing it with the reduction function r_s correspondent to the chosen s to obtain the *reduced symbol*

$$u_s := r_s(g), \tag{1}$$

that is encoded with $\log_2|K_s| < \log_2|G|$ bits and sent to the other peers.

A node that wants to recover the original content symbol g contacts R peers, receives the corresponding reduced versions $u_{s_1}, u_{s_2}, \ldots, u_{s_R}$ and recovers g by solving the system

$$\begin{aligned} u_{s_1} &= r_{s_1}(g), \\ u_{s_2} &= r_{s_2}(g), \\ &\vdots \\ u_{s_R} &= r_{s_R}(g). \end{aligned} \tag{2}$$

Intuitively, if R is large enough, the peer can recover g. A key concept that we will use in this paper is the concept of a reduction scheme that it is *R-recoverable*. Informally, a scheme is R-recoverable if every $g \in G$ can be recovered by the knowledge of any set of R different reduced versions.

Definition 2 (R-recoverable). Let $\{r_s : G \to K_s, s \in S\}$ be a reduction scheme and define, for every set of R different reduction parameters $s_1, \ldots, s_R \in S$, the function $\phi_{s_1,\ldots,s_R} : G \to K_{s_1} \times \cdots \times K_{s_R}$ as

$$\phi_{s_1,\ldots,s_R}(g) := [r_{s_1}(g), \ldots, r_{s_R}(g)]. \tag{3}$$

The reduction scheme will be said to be R-recoverable if for every choice of R different parameters $s_1, \ldots, s_R \in S$, function $\phi_{s_1,\ldots,s_R}$ is injective.

The reduction scheme will be said to be R-tight if it is R-recoverable and for every choice of $R-1$ reduction parameters $s_1, \ldots, s_{R-1} \in S$, the corresponding function $\phi_{s_1,\ldots,s_{R-1}}$ is *not* injective.

Remark 2. Note that in Definition 2, we require only $\phi_{s_1,\ldots,s_R}$ to be *injective*, not *bijective*. That is, we *do not* require that system (2) have a solution for every choice of $u_{s_i} \in K_{s_i}$, $i = 1, \ldots, R$ (it could have none), but we require that *if a solution exists, then it is unique.*

The property of being R-recoverable is very interesting for applicative purposes, since it allows each node to choose its parameter s at random while granting (with large probability) the possibility of recovering the content g, since the probability of having two nodes choosing the same value can be made as small as desired by choosing $|S|$ large enough. A reduction scheme that is R-recoverable has also other interesting characteristics such as being resilient to data loss (if the node contacts $N > R$ peer, it can recover g as soon as it receives R reduced versions out of N), counteracting poisoning [7] (the node uses R reduced versions to recover g and uses the remaining $N - R$ to check for the correctness of the result [1]), and reducing jitter [10].

Example 1 (Vandermonde reduction scheme). In order to give a concrete example of the just described abstract model, it is worthwhile to show how the reduction approach in [1] can be adapted to the described setup. The approach in [1] maps a block of Rd bits of the content stream in a column vector

$$\mathbf{c} = [a_1, \ldots, a_R]^t, \tag{4}$$

where each a_i belongs to the Galois field with 2^d elements $\mathbb{F}_{2^d}$. Each node chooses at start-up time an element $s \in \mathbb{F}_{2^d}$ and constructs the row vector

$$\mathbf{r}_s := \left[1, s, \ldots, s^{R-1}\right]. \tag{5}$$

In order to produce the reduced version of vector $\mathbf{c}$, the node multiplies $\mathbf{c}$ by $\mathbf{r}_s$ to obtain $u_s = \mathbf{r}_s\mathbf{c}$. Value u_s is sent to the other peers, and its transmission requires only d bits instead of Rd bits. Therefore, the required upload bandwidth is R times smaller.

In order to recover $\mathbf{c}$, a node can ask for R different values $u_{s_1}, \ldots, u_{s_R}$ and solve the linear system

$$\begin{bmatrix} u_{s1} \\ u_{s2} \\ \cdots \\ u_{sR} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & s_1 & \cdots & s_1^{R-1} \\ 1 & s_2 & \cdots & s_2^{R-1} \\ \vdots & \vdots & & \vdots \\ 1 & s_R & \cdots & s_R^{R-1} \end{bmatrix}}_{\mathbf{R}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_R \end{bmatrix}}_{\mathbf{c}}. \tag{6}$$

Note that matrix $\mathbf{R}$ in (6) is a Vandermonde matrix, and it is invertible as soon as all the s_i values are different.

Reformulated with the language of the formalization presented here, we can say that the content alphabet is the set of R-dimensional vectors with entries in $\mathbb{F}_{2^d}$; that is, $G = \mathbb{F}_{2^d}^R$; the reduction functions are parametrized by $s \in S = \mathbb{F}_{2^d}$ and are defined as

$$r_s(\mathbf{c}) := \left[1, s, \ldots, s^{R-1}\right]\mathbf{c}, \tag{7}$$

and, finally, $K_s = \mathbb{F}_{2^d}$ for every $s \in S$.

Note that this reduction scheme is R-tight.

2.2. Group-Based Reduction Procedures. The setup described in Section 2.1 is very general. In order to simplify the study, it is worthwhile to restrict the model above by adding to it some structure. A structure that is quite powerful but still quite general to be applied in several cases of practical interest is the structure of *group*. In this paper, we need only basic notions and results of group theory. For the sake of completeness, Appendix A summarizes the concepts used in this paper, and a more detailed description can be found in the literature [11, 12].

In the following we will denote the group operation as a product and will use the symbol e for the neutral element of a group. We will denote with $E = \{e\}$ the trivial group that contains only the neutral element. Group isomorphism will be denoted with $\simeq$. The ring of integers modulo N will be denoted as $\mathbb{Z}/N\mathbb{Z}$.

Definition 3. A group-based reduction scheme (GBRS) is a reduction scheme $\{r_s : G \to K_s, s \in S\}$, where *the content alphabet* G and each reduced alphabet K_s, $s \in S$ are finite groups and each r_s is a group epimorphism (that is, a surjective homomorphism). Note that there is no loss of generality in requiring that each r_s is an epimorphism, since one can always replace K_s with $\mathrm{Im}(r_s)$.

Remark 3. Note that the reduction scheme presented in Example 1 is a group-based reduction scheme, since $\mathbb{F}_{2^d}^R$ and $\mathbb{F}_{2^d}$ are groups (with respect to the sum) and maps $r_s(\mathbf{c}) = \mathbf{r}_s\mathbf{c}$ are clearly group homomorphisms. We will see in Section 4 examples of group-based schemes that are not based on a vector space structure.

2.2.1. Normalized Form. According to Definition 3, in order to specify a group-based scheme, one must specify the content group G, the reduced groups K_s, and the epimorphisms r_s. These requirements can be simplified by exploiting the fundamental homomorphism theorem [11] that implies that r_s can be written as

$$r_s(x) = \eta\left(\pi_{\ker(r_s)}(x)\right), \tag{8}$$

where $\pi_{\ker(r_s)} : G \to G/\ker(r_s)$ is the natural map associated with $G/\ker(r_s)$ [11] and $\eta : G/\ker(r_s) \to \mathrm{Im}(r_s) = K_s$ is an isomorphism. Since isomorphic groups are basically the same group, isomorphism η does not have any practical consequences in our context, so we can restrict ourselves to the case where reduced alphabets are quotient groups G/H, where H is a normal subgroup of G (the set of normal subgroups coincides with the set of subgroups that are kernel of some homomorphism [11]) and map $r : G \to G/H$ is the natural map of G in G/H (i.e., $r(x) = xH$, where xH is the coset of G/H to which x belongs) [11].

Definition 4 (GBRS normalized form). A group-based reduction scheme $\{r_s : G \to K_s, s \in S\}$ is said to be in normalized *form* if for every $s \in S$

(1) $K_s = G/H_s$ for some $H_s \lhd G$,

(2) map r_s is the natural map $\pi_{H_s} : G \to G/H_s$ associated with G/H_s.

Observe that in order to specify a GBRS in normalized form, it suffices to specify a set $\{H_1, H_2, \ldots, H_L\}$ of normal subgroups of G. With a minor abuse of language, we will use the term *reduction scheme* also for set $\{H_1, H_2, \ldots, H_L\}$.

With the normalized form, the reduced version of a content symbol g is always a coset gH which can be considered as "g reduced modulo H." The group H represents the "uncertainty" that one has about g when it knows its reduced version gH: the smaller the cardinality of H, the smaller the uncertainty about g. If $H = E$, no uncertainty is present, and g is exactly known.

Example 2. It is worthwhile to describe the Vandermonde reduction scheme of Example 1 as a GBRS in normalized form. The kernel of map $r_s(\mathbf{u}) = \mathbf{r}_s\mathbf{u}$ is the subspace of $\mathbb{F}_{2^d}^R$ orthogonal to $\mathbf{r}_s$. The elements of the quotient group $\mathbb{F}_{2^d}^R/\ker(r_s)$ are translated versions of the subspace orthogonal to $\mathbf{r}_s$. Note that every coset U of $\mathbb{F}_{2^d}^R/\ker(r_s)$ is uniquely identified by the value of the product $\mathbf{r}_s\mathbf{u}$, where $\mathbf{u}$ is any element of coset U (it is easy to see that the product does not depend on the chosen $\mathbf{u}$).

Summarizing, value $r_s(\mathbf{u})$ can be computed as follows: first $\mathbf{u}$ is mapped to the coset $\mathbf{u} + \ker(r_s)$ to which it belongs, then any representative of the coset is left multiplied by $\mathbf{r}_s$. It is easy to verify that the latter step is an isomorphism from

$\mathbb{F}_{2^d}^R/\ker(r_s)$ to $\mathbb{F}_{2^d}$. Therefore, in order to study the theoretical properties of the Vandermonde reduction scheme, one can replace each $\mathbf{r}_s$ with the natural map $\pi_{\ker(r_s)}$.

3. Properties of GBRS

In this section, we will study some properties of GBRS. First, in Section 3.1, we will derive conditions for the existence of a solution of system (2), and we will show that depending on the choice of the subgroups, system (2) could have no solution for some choice of the values u_{s_i}. If a reduction scheme is such that some R-ple of reduced values is not admissible, one expects that the R-ple of reduced values that can be obtained has some redundancy within. This idea is further pursued in Section 3.1.2, where it is also shown that the redundancy can be used to counteract stream poisoning attacks [7].

The Vandermonde reduction scheme described in Example 1 is deeply linked with the secret sharing scheme of [13]. Actually, the secret sharing scheme of Shamir can be derived from the Vandermonde scheme by replacing some information data with random data. This idea is discussed in greater detail in Section 3.2, where it is shown how nonredundant reduction schemes (i.e., schemes such that (2) has always a solution for every R-ple of reduced values) can be easily converted in secret sharing schemes.

3.1. Reconstruction. As explained in Section 2.1, recovering the original content requires to solve system (2). In this section, we are going to study the reconstruction problem by showing that given two reduced versions uH and vK, one can combine them in order to get a "virtual" reduced version $w(H \cap K)$, reduced with respect to the smaller uncertainty group $H \cap K$. Intuitively, by combining the virtual reduced version with other reduced versions, one can make the uncertainty group smaller and smaller until the original content symbol is recovered. In some sense, this corresponds to solving system (2) by means of an iterative approach: first, we determine the set of values of g that satisfy the first two equations of (2), then we use the third equation to refine the solution, and so on until only one solution remains. The problem to be solved at the first step of the iterative algorithm can be formalized as follows.

Problem 1. Let H and K be normal subgroups of G, and let $uH \in G/H$ and $vK \in G/K$. Find all the $a \in G$ such that $\pi_H(a) = uH$ and $\pi_K(a) = vK$ (or, equivalently, find the intersection $uH \cap vK$).

The following property gives an answer to Problem 1.

Property 1. Let H and K be normal subgroups of G, and let $uH \in G/H$ and $vK \in G/K$. Let

$$\mathcal{S} := \{g \in G : \pi_H(g) = uH,\ \pi_K(g) = vK\} \tag{9}$$

be the set of content symbols $g \in G$ that have uH and vK as reduced versions.

(1) Set $\mathcal{S}$ is not empty if and only if

$$u^{-1}v \in HK, \tag{10}$$

or, equivalently, $uHK = vHK \in G/HK$. Note that since $H \subset HK, \ldots$ chosen to represent uH.

(2) if (10) is satisfied, set $\mathcal{S}$ can be written as

$$\mathcal{S} = a(H \cap K) \in \frac{G}{(H \cap K)}, \tag{11}$$

where a is any element of $\mathcal{S}$. In other words, $\mathcal{S}$ is a coset of $G/(H \cap K)$.

Proof.

Step 1 (If $\mathcal{S} \neq \emptyset$, then condition (10) holds). Let $g \in \mathcal{S}$. Since $\pi_H(g) = uH$ and $\pi_K(g) = vK$, there must exist $h \in H$ and $k \in K$ such that

$$uh = g = vk, \tag{12}$$

that implies $u^{-1}v = hk^{-1} \in HK$, that is, (10).

Step 2 (If condition (10) holds, then $\mathcal{S} \neq \emptyset$). If $u^{-1}v \in HK$, one can find $h \in H$ and $k \in K$ such that $u^{-1}v = hk$. It follows that $uh = vk^{-1}$. Since $uh \in uH$ and $vk^{-1} \in vK$, it follows that $uh \in \mathcal{S}$. Incidentally, note that if one knows how to decompose an element of HK into a product of an element of H and an element of K, this procedure allows to find a solution in $\mathcal{S}$.

Step 3 (If $\mathcal{S} \neq \emptyset$, then (11) holds). Define homomorphism $\phi : G \to G/H \times G/K$ as $\phi(g) = (\pi_H(g), \pi_K(g))$ and observe that $\mathcal{S} = \phi^{-1}(uH, vK)$, that is, $\mathcal{S}$ is the inverse image of (uH, vK). Since ϕ is an homomorphism, it is known that (if $\mathcal{S} \neq \emptyset$) $\phi^{-1}(uH, vK)$ is a coset of $G/\ker(\phi)$. The thesis will follow if one can prove that

$$\ker(\phi) = H \cap K. \tag{13}$$

Equation (13) can be proved by observing that $a \in \ker(\phi)$ if and only if $\pi_H(a) = H$ (i.e., equivalent to $a \in H$) and $\pi_K(a) = K$ (i.e., equivalent to $a \in K$). □

Several remarks about Property 1 are in order.

Remarks. (1) Suppose system (2) has solution $\hat{g}$. Clearly, $\hat{g}$ must necessarily belong to set (11) so that set (11) can be written as $\hat{g}(H \cap K)$. It follows that (11) is the *reduced version of the solution* $\hat{g}$ *with respect to group* $H \cap K$. This implies that Property 1 can be applied to every step of the iterative algorithm outlined at the beginning of this section.

Note that the availability of an iterative algorithm that solves system (2) one equation at time can be interesting from an implementation point of view, since it allows to spread the computational burden over the time, updating the solution as soon as new data are received. Depending on the applicative context, this can be more convenient than waiting for all data to arrive before starting the reconstruction.

(2) Note that condition (10) poses a compatibility condition on the pair (uH, vK). Such a condition can,

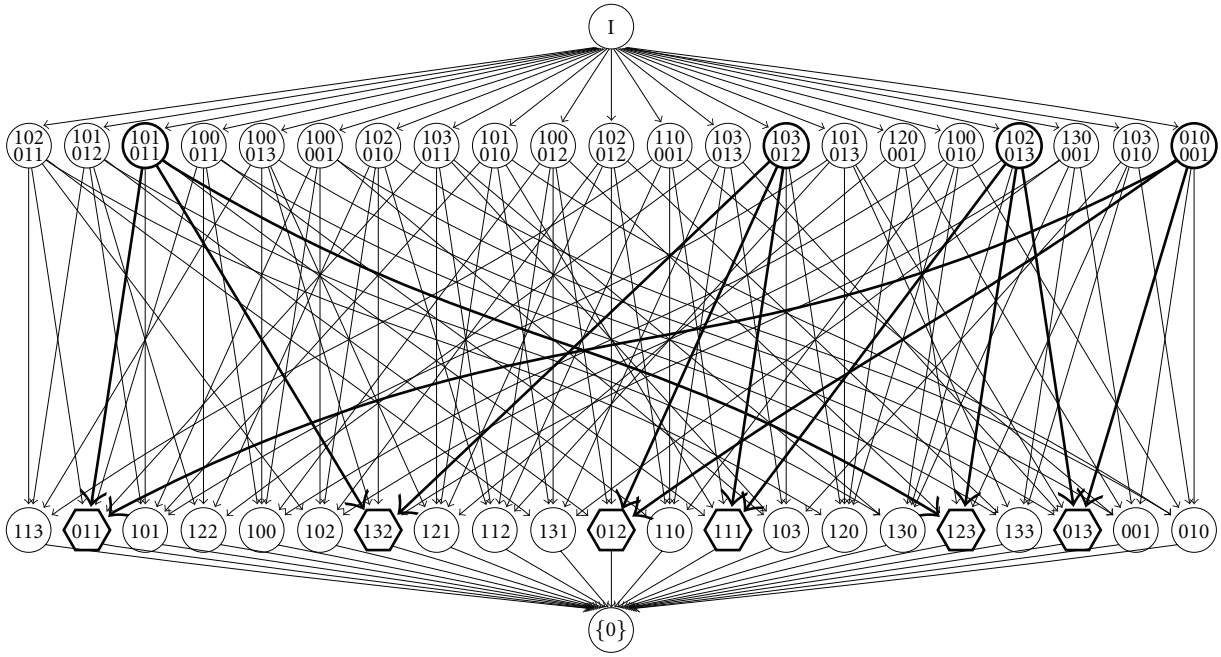

Figure 1: Lattice of the subspaces of $\mathbb{F}_4^3$.

however, be trivially true if HK is equal to the whole group G. If HK is a proper subset of G, not every pair (uH, vK) is admissible, and this, intuitively, implies that there is some redundancy in the pair (uH, vK). This aspect will be discussed more in detail in Section 3.1.2.

3.1.1. The Reconstruction Problem and the Lattice of Normal Subgroups. Property 1 has a nice interpretation in the context of the lattice of the normal subgroups of G (see Appendix B for a brief summary about lattices and [11] for a more detailed exposition). According to Property 1, the uncertainty group $H \cap K$ of the combined version is the greatest lower bound $H \wedge K$ of H and K (i.e., their first common descendant on the lattice graph), while the group associated with constraint (10) is the smallest upper bound $H \vee K$ (i.e., their first common ancestor on the lattice graph).

Example 3. Figure 1 shows the lattice graph of the subspaces of $\mathbb{F}_4^3$, together with the subspaces involved in the Vandermonde reduction scheme. In the case of Figurge 1, $\mathbb{F}_4$ is implemented as the polynomials with coefficients in $\mathbb{Z}/2\mathbb{Z}$ modulo $z^2 + z + 1$.

Each node in Figure 1 is labeled with a basis of the space, and the elements of $\mathbb{F}_4$ are represented as integer numbers in $\{0, 1, 2, 3\}$ whose digits in the binary representation are the coefficients of the corresponding element of $\mathbb{F}_4$ (e.g., 3 corresponds to $z+1$). The top node, labeled with I, represents the whole space $\mathbb{F}_4^3$, while the bottom node, labeled with $\{0\}$, represents the trivial space.

As explained in Example 2, the groups associated with the Vandermonde scheme are the $R-1$-dimensional subspaces orthogonal to vectors of type (5). In Figure 1, these vector spaces are 2-dimensional and correspond to the four nodes marked with a bold circle. In order to obtain the intersection of two of the spaces associated with the Vandermonde scheme, one needs to find the first common descendant of the two spaces. By considering all the six different unordered pairs of spaces, one obtains the six one-dimensional spaces marked with bold hexagons in Figure 1.

Note that any triple of spaces has as common descendant the trivial space $\{0\}$, coherently with the fact that the scheme is 3-recoverable. Moreover, any pair of spaces has as common ancestor, the whole space $\mathbb{F}_4^3$, coherently with the fact that system (6) is solvable for any vector $[u_{b_1}, \ldots, u_{b_R}]$ of reduced values.

3.1.2. Redundancy in a GBRS. It is worthwhile commenting about the meaning of constraint (10) in the context of network coding for peer-to-peer streaming. Remember that uH and vK represent two reduced versions received from two peers. According to Property 1, if uH and vK have been obtained by reducing the same content symbol g, then uH and vK are "compatible" according to (10). If $HK \neq G$, constraint (10) is not trivial, and not all the pairs (uH, vK) are valid.

Intuitively, this is very similar to the case when redundant bits are added to protect communications from errors. Actually, adding redundant bits to the information to be transmitted constrains the set of admissible sequences of bits, and if the received sequence does not satisfy the constraints induced by the redundant bits, the receiver can deduce that an error occurred. Here, similarly, if uH and vK do not satisfy (10), we can deduce that at least one reduced value must be incorrect. In Section 3.1.3, it is shown how it is possible to exploit this possibility to counteract poisoning attacks when network coding is used for streaming over peer-to-peer networks.

The idea that if $HK \neq G$, then some redundancy is present, is confirmed by the following result.

Property 2. Let the notation be as in Property 1. The following equality holds:

$$\left|\frac{G}{H\cap K}\right| = \left|\frac{G}{H}\right|\left|\frac{G}{K}\right|\left|\frac{G}{HK}\right|^{-1}. \tag{14}$$

Proof. Since HK/K is isomorphic to $H/(H\cap K)$ [11], it follows that

$$\frac{|HK|}{|K|} = \frac{|H|}{|H\cap K|}. \tag{15}$$

By exploiting (15), one can write

$$\begin{aligned}\left|\frac{G}{H\cap K}\right| = \frac{|G|}{|H\cap K|} &= \frac{|HK||G|}{|H||K|}\\ &= \frac{|G|}{|H|}\frac{|G|}{|K|}\frac{|HK|}{|G|}\\ &= \left|\frac{G}{H}\right|\left|\frac{G}{K}\right|\left|\frac{G}{HK}\right|^{-1}.\end{aligned} \tag{16}$$

□

By taking the logarithms of (14) and reorganizing, one obtains

$$\log_2\left|\frac{G}{HK}\right| = \left(\log_2\left|\frac{G}{H}\right| + \log_2\left|\frac{G}{K}\right|\right) - \log_2\left|\frac{G}{H\cap K}\right|. \tag{17}$$

Observe that the sum in parenthesis represents the number of bits that we used to receive the two reduced versions, while the last term on the right-hand side of (17) can be interpreted as the number of bits necessary to describe the result of the combination of the two reduced versions. It follows that their difference can be interpreted as the "redundancy" of the system in the sense that it is the difference between the number of bits spent and the number of bits that we got after the combination. Note that if $HK = G$ (the case when condition (10) is always verified), then the last term of (17) is zero; that is, no redundancy is added.

Definition 5. Scheme $\{H_1, H_2, \ldots H_K\}$ will be said to be nonredundant if for every choice of different H_{j_n}, $n = 0, \ldots, L$ such that $H_{j_1}\cap\cdots\cap H_{j_L} \neq E$, then $H_{j_0}(H_{j_1}\cap\cdots\cap H_{j_L}) = G$.

Example 4. Consider the case of the Vandermonde reduction scheme. In this case, H and K are two spaces, of dimension $R-1$, orthogonal, respectively, to vectors $\mathbf{r}_1$ and $\mathbf{r}_2$. The intersection $H\cap K$ is the space of the vectors that are orthogonal to *both* $\mathbf{r}_1$ and $\mathbf{r}_2$, and, as known, it has dimension $R-2$. In the Vandermonde scheme, product HK is the vector space sum $\mathbf{r}_1^{\perp} + \mathbf{r}_2^{\perp}$. Since spaces $\mathbf{r}_1^{\perp}$ and $\mathbf{r}_2^{\perp}$ have dimension $R-1$, their sum has dimension R (that corresponds to the no redundancy case $HK = G$) unless the two spaces coincide. Therefore, the Vandermonde scheme has no redundancy.

We will introduce in the following a reduction scheme based on the Chinese remainder theorem (CRT) that allows for the introduction of redundancy.

3.1.3. Counteracting Stream Poisoning. One important security threat in P2P streaming is the *stream poisoning attack* where a node sends wrong packets on the P2P network with the objective of disrupting the communication [7]. A reduction-based approach can help counteracting this attack. The idea is very simple: if the reduction scheme is R-recoverable, a node asks data to $N > R$ peers, uses R reduced versions to recover the content, and then uses the remaining $N-R$ reduced versions to check the correctness of the result; by the knowledge of which tests fail, it is possible to spot who tried the attack [1] (if all the tests fail, it means that a corrupted value was used in the reconstruction process; the node can retry the tests using a different subset of R in the reconstruction step). It is possible to show that this test is robust against a coordinated attack of at most $N-R$ peers [1].

A drawback of the test above is that one needs to recover first the content symbol and then do the test. If, by chance, a corrupted value is used in the reconstruction process, the node needs to try the reconstruction again. It would be more efficient if the node was able to spot the corrupted data *before* doing the reconstruction process.

This can be done by using a redundant scheme and exploiting Property 1 by checking (10) before attempting the reconstruction.

3.2. Generalized Secret Sharing. Secret-sharing techniques allow one to share a secret among N people with the constraint that (i) R people, putting their information together, can recover the secret and (ii) $R-1$ people that put their information together cannot deduce *anything* about the secret [8, 13]. We will say that the scheme achieves *perfect secrecy*. Note that secret sharing is a problem very similar to the reduction problem described in this paper but with the additional constraint of (ii).

Actually, the Vandermonde reduction scheme described in Example 1 can be easily converted into the secret-sharing scheme described in [8, 13]. More precisely, suppose that the secret to be shared is represented by a value $x \in \mathbb{F}_{2^d}$. The scheme of [13] builds vector $\mathbf{c}$ in (4) by setting $a_1 = x$ and choosing $a_2, \ldots, a_R$ at random. Successively, N reduced values are created and distributed among the participants. In order to recover the secret, one collects R reduced versions, gets $\mathbf{c}$ by solving (6), and takes the first component of the result. Note that taking the first component of $\mathbf{c}$ is equivalent to right-multiplying $\mathbf{c}$ by $\mathbf{e}_1^t = [1, 0, \ldots, 0]$.

As said above, secret sharing has the additional constraint of perfect secrecy; that is, any set of $R-1$ participants cannot deduce anything about the secret. This can be easily verified by observing that $\mathbf{e}_1$ does not belong to the space generated by any set of $R-1$ vectors of type $\mathbf{r}_s$, $s \neq 0$ (note that $\mathbf{e}_1 = \mathbf{r}_0$), and this implies that from the knowledge of $R-1$ reduced values, nothing can be inferred about the value of $x = a_1 = \mathbf{e}_1\mathbf{c}$.

In this section we will show how the procedure used to convert the Vandermonde scheme into a secret sharing scheme can be generalized to any GBRS. Observe that the above described secret sharing procedure can be reformulated as follows: map the information to be shared x into

a vector $\mathbf{v} = x\mathbf{e}_1$ belonging to the one-dimensional space $V_1 := \text{span}\{\mathbf{e}_1\}$ generated by $\mathbf{e}_1$, successively add to $\mathbf{v}$ a random vector $\mathbf{q}$ belonging to the $R-1$-dimensional space $\mathbf{e}_1^{\perp}$ orthogonal to $\mathbf{e}_1$, and finally vector $\mathbf{v}+\mathbf{q}$ is processed with the reduction scheme. The fact that the intersection between V_1 and $\mathbf{e}_1^{\perp}$ is trivial allows one to recover uniquely $\mathbf{v}$ (and x) from $\mathbf{v}+\mathbf{q}$.

In order to extend the secrete sharing scheme in the GBRS context, we need an *obfuscating subgroup* $M \lhd G$ that will play the role of $\mathbf{e}_1^{\perp}$.

The generalized secret sharing scheme is the following: let I be any set of representatives of G/M; we encode the information to be shared as an element $v \in I$, we draw at random $q \in M$ and compute $h = vq \in vM$, then we apply the reduction scheme to h. After recovering h, one can obtain v by applying π_M, the natural map associated with M, to h. The only thing that remains to be checked is to verify when this scheme achieves perfect secrecy. Remember that a reduced value is a coset of a quotient group G/H, where H is a normal subgroup of G. Let uH be the reduced version of $h = vq$, with q a random element of M. Our objective is to deduce some information about v from the knowledge of uH. We will say that a value $\ell \in G$ is *compatible with* uH if there are $m \in M$ and $k \in H$ such that $\ell m = uk$, or, alternatively,

$$\ell \in uHM. \tag{18}$$

Note that if ℓ is *not* compatible with uH, then we obtain some information about the secret value, since we know that the secret *cannot* be ℓ. Therefore, the scheme will achieve *perfect secrecy with reduction group* H if every $\ell \in I$ is compatible with uH for every $u \in G$.

Remark 4. From (18), it follows at once that ℓ is compatible with uH if and only if any other element of ℓM is compatible with uH. This implies that one can change the set of representatives I without changing the secrecy characteristics of the scheme.

Definition 6. Let H and M be normal subgroups of G. We say that perfect secrecy is achieved if for every $u \in G$, every $\ell \in I$ is compatible with uH.

Property 3. Let H and M be normal subgroups of G. Perfect secrecy is achieved if and only if $HM = G$.

Proof. If $HM = G$, then perfect secrecy is achieved. If $HM = G$, it is obvious that $\ell \in uHM = uG = G$ for every $\ell \in I$.

If $HM \neq G$, then perfect secrecy is not achieved. Suppose now that $HM \neq G$; that is, there is $o \in G$ that does not belong to HM. Observe that according to Remark 4, we can suppose without loss of generality, $e \in I$; we will prove that e is not compatible with o^{-1}. Indeed, if e was compatible with o, (18) would imply $o \in HM$. Therefore, e is not compatible with o^{-1}, and perfect secrecy is not achieved. □

If condition $HM = G$ is fulfilled, it is possible to prove that perfect secrecy is achieved even in a stronger, information theoretical sense. Indeed, although Property 3 claims that if $HM = G$, then any uH can be obtained from any information symbol $\ell \in I$, it could happen that the probability of obtaining uH from ℓ could depend on ℓ. In this case, an attacker could deduce from uH something about ℓ. The following result shows that this is not the case.

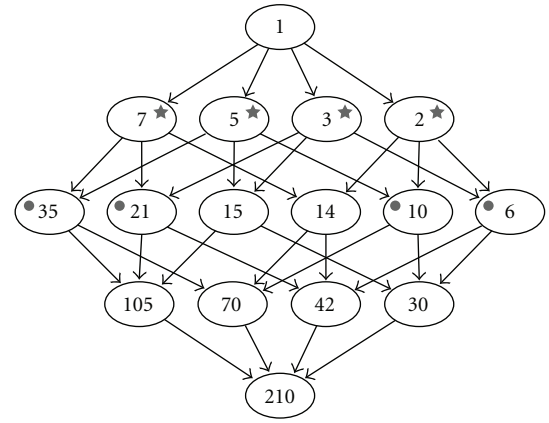

Figure 2: Lattice of the subgroups of $\mathbb{Z}/210\mathbb{Z}$. A node labeled with M represents group $M\mathbb{Z}/210\mathbb{Z}$.

Property 4. Let L be a random variable assuming values in I, and let Q be a random variables assuming values in M and uniformly distributed. If $HM = G$, then for every $\alpha \in G$,

$$P[\pi_H(LQ) = \alpha H \mid L = \ell] = \frac{|H \cap M|}{|M|}. \tag{19}$$

According to Property 4, $\pi_H(LQ)$, the reduced version of LQ, is statistically independent on L, and this implies that mutual information $I(\pi_H(LQ); L)$ [14] is zero, so that an attacker cannot deduce anything about L from $\pi_H(LQ)$.

The proof of Property 4 is simplified by using the following lemma.

Lemma 1. *Let $\beta \in G$. If $HM = G$, cardinality $C_\beta := |(\beta H) \cap M|$ does not depend on β.*

Proof.

Step 1 (If $h \in H$, then $C_\beta = C_{\beta h}$). This follows at once by observing that $\beta H = \beta h H$.

Step 2 (*If $m \in M$, then $C_{m\beta} = C_\beta$*). This follows at once by observing that $m[(\beta H) \cap M] = (m\beta H) \cap mM = (m\beta H) \cap M$.

Step 3 ($C_\beta = C_e$ *for every* $\beta \in G$). Since both M and 0 are normal subgroups of G, $MH = HM = G$. It follows that every $\beta \in G$ can be written as $\beta = mh$ with $m \in M$ and $h \in H$. By exploiting Steps 1 and 2 above, one deduces that

$$C_\beta = C_{mh} = C_h = C_e. \tag{20}$$

□

Proof of Property 4. It is a simple verification

$$\begin{aligned}
&P[\pi_H(LQ) = \alpha H \mid L = \ell] \\
&\quad = P[LQ \in \alpha H \mid L = \ell] \quad \text{By definition of } \pi_H \\
&\quad = P[\ell Q \in \alpha H \mid L = \ell] \\
&\quad = P[Q \in \ell^{-1}\alpha H \mid L = \ell] \\
&\quad = P[Q \in \ell^{-1}\alpha H] \quad \text{Independence of } L \text{ and } Q \\
&\quad = P[Q \in (\ell^{-1}\alpha H) \cap M] \quad \text{Since } Q \in M \\
&\quad = \frac{|(\ell^{-1}\alpha H) \cap M|}{|M|} \quad Q \text{ uniformly distributed} \\
&\quad = \frac{|H \cap M|}{|M|} \quad \text{Lemma 1.}
\end{aligned} \tag{21}$$

□

One can interpret Property 3 by saying that in order to have perfect secrecy, one must choose M large enough so that, when combined with any H that can result from the combining process, it generates the whole group G. This is what is done in the secret sharing scheme based on the Vandermonde scheme. In this case, M is a space of dimension $R-1$ that when combined with any nontrivial space resulting from the Vandermonde scheme, it generates the whole space $\mathbb{F}_{2^d}^R$.

Property 5. Let $\{H_0, H_1, H_2, \ldots H_K\}$ be a nonredundant R-recoverable reduction scheme with $R \leq K$. Obfuscating group $M = H_0$ together with reduction scheme $\{H_1, H_2, \ldots H_K\}$ is a secret-sharing scheme that achieves perfect secrecy.

Proof. Since scheme $\{H_0, H_1, \ldots H_K\}$ is not redundant, then we have, by definition, $H_0(H_{j_1} \cap \cdots \cap H_{j_L}) = G$ as soon as $H_{j_1} \cap \cdots \cap H_{j_L} \neq E$. □

According to Property 5, constructing secret-sharing schemes that achieve perfect secrecy is simple as soon as one has a nonredundant scheme with enough subgroups: just use one of the subgroups as the obfuscating group.

4. Examples of Alternative Reduction Schemes

In order to show the flexibility and the generality of the presented theory, in this section, we present some schemes that are not expressible as schemes based on vector spaces over finite fields. With respect to the Vandermonde scheme, the schemes presented here have the characteristic that they do not require arithmetic in finite fields but only ordinary integer arithmetic, and this, depending on the applicative context, could be interesting from a complexity point of view. Moreover, the redundancy of the schemes presented here can be easily adapted to the specific application requirements: low redundancy (or none at all) if efficiency is required and more redundancy if error protection is needed.

4.1. CRT-Based Reduction Scheme. Let $p_1, p_2, \ldots, p_L$ be mutually prime numbers (i.e., the greatest common divisor of p_i and $p_j \neq p_i$ is 1), and let $N = p_1 \cdots p_L$. We will consider the group $\mathbb{Z}/N\mathbb{Z}$ of the integers modulo N. Every subgroup of $\mathbb{Z}/N\mathbb{Z}$ has the form $M\mathbb{Z}/N\mathbb{Z}$, where M divides N. The reduction of $x = u + N\mathbb{Z} \in \mathbb{Z}/N\mathbb{Z}$ with subgroup $M\mathbb{Z}/N\mathbb{Z}$ is $(u \bmod M) + M\mathbb{Z}/N\mathbb{Z}$. Note that since M divides N, coset $(u \bmod M) + M\mathbb{Z}/N\mathbb{Z}$ does not depend on the representative chosen for $u + N\mathbb{Z}$.

Let us consider, as an example, the case $N = 210$. In this case, the content symbols are integers in the range $0 \cdots 209$, the reduction with subgroup $M\mathbb{Z}/210\mathbb{Z}$ is the usual reduction modulo M, and Property 1 reduces itself to the Chinese remainder theorem. A reduction scheme based on the group $\mathbb{Z}/210\mathbb{Z}$ is uniquely specified by giving a set of subgroups of $\mathbb{Z}/210\mathbb{Z}$.

Figure 2 shows the lattice of the subgroups of $\mathbb{Z}/210\mathbb{Z}$. Subgroup $M\mathbb{Z}/210\mathbb{Z}$ is labeled with M in Figure 2; therefore, the bottom node of Figure 2 represents the trivial group $E \simeq 210\mathbb{Z}/210\mathbb{Z}$, while the top node represents the content alphabet $G = \mathbb{Z}/210\mathbb{Z}$. By choosing the nodes marked with a star, one obtains a reduction scheme with no redundancy that enjoys the 4-reconstruction (tight) property, while choosing the node marked with a circle, one obtains a scheme that enjoys a 3-reconstruction property but not tight, since in some cases, only two reduced versions suffice. Moreover, the scheme associated with the circles is redundant, since the least upper bound of two nodes marked with circles is not the top node.

4.2. Point Lattice Reduction Schemes. Let $\mathbf{M} \in \mathbb{Z}^{D\times D}$ be a square matrix with integer entries with $\det \mathbf{M} \neq 0$. The *point lattice* of base $\mathbf{M}$ is the set $\mathbf{M}\mathbb{Z}^D \subset \mathbb{R}^D$ obtained by taking *integer* linear combinations of the columns of $\mathbf{M}$; that is,

$$\mathbf{M}\mathbb{Z}^D := \left\{\mathbf{Mn}, \mathbf{n} \in \mathbb{Z}^D\right\}. \tag{22}$$

(Typically, $\mathbf{M}\mathbb{Z}^D$ is called simply *lattice*; here we use the term *point lattice* in order to avoid confusion with the lattices introduced in Appendix B.) A point lattice is clearly a subgroup of $\mathbb{Z}^D$. It is known [11, 15] that $|\mathbb{Z}^D/\mathbf{M}\mathbb{Z}^D| = \det \mathbf{M}$. Since $\mathbb{Z}^D/\mathbf{M}\mathbb{Z}^D$ is finite, it is a suitable group for building reduction schemes.

Consider, for example, the case where $\mathbf{M} = \mathrm{diag}(4, 4)$. It is easy to see that each class of $\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2$ can be uniquely identified by its representative belonging to the set $\{0, \ldots 3\} \times \{0, \ldots, 3\}$. Such a representative can be encoded by using four bits: two bits per component. Note that each subgroup of $\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2$ has the form $\mathbf{N}\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2$, where $\mathbf{N}$ is an integer matrix such that $\mathbf{N}^{-1}\mathbf{M}$ has integer entries [15]. By exploiting the Hermite normal form theorem, it is possible to show that $\mathbf{N}$ can be supposed without loss of generality in lower triangular form. If n_1 and n_2 are the diagonal elements of $\mathbf{N}$, it is easy to check that every class of $[\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2]/[\mathbf{N}\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2]$ can be uniquely identified by its representative belonging to the set $\{0, \ldots, n_1 - 1\} \times \{0, \ldots, n_2 - 1\}$. Since $\det \mathbf{N} = n_1 n_2$ must divide $\det \mathbf{M} = 16$, we are granted that both n_1 and n_2 must be powers of two, making the binary representation of the representative trivial.

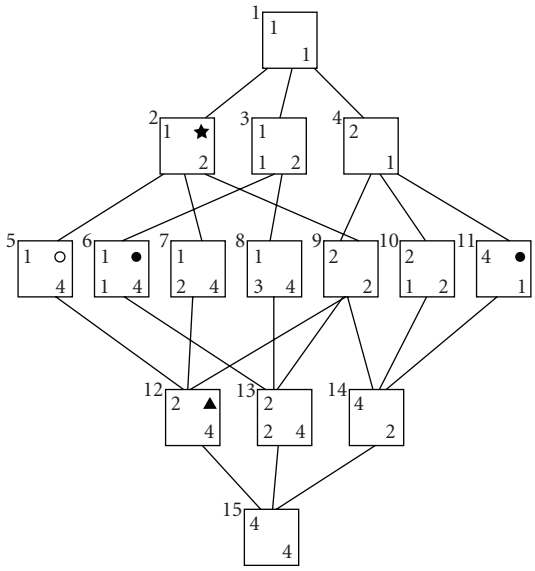

Figure 3: Lattice of the subgroups of $\mathbb{Z}^2/4\mathbb{Z}^2$. A node labeled with **M** represents group $\mathbf{M}\mathbb{Z}^2/4\mathbb{Z}^2$.

Figure 3 shows the lattice graph of the subgroups of $\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2$. (The lattice graph of Figure 3 can be obtained by using the algorithm in [16].) As for the case of Figure 2, each node in Figure 3 is labeled with the corresponding **N** matrix; therefore, the top node corresponds to the content alphabet $\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2$, while the bottom node corresponds to the trivial group $E \simeq \mathbf{M}\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2$.

Several choices for reduction schemes are possible.

(1) Use the two subgroups marked with full circles and the one marked with an empty circle. This gives a 2-tight scheme (the first common descendant of each pair of nodes is the node corresponding to $E \simeq \mathbf{M}\mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2$) without redundancy (the first common ancestor of each pair of nodes is the node corresponding to $G = \mathbb{Z}^2/\mathbf{M}\mathbb{Z}^2$).

If $[x, y]^t$, $x, y \in \{0, \ldots, 3\}$ represents the symbol to be reduced, it is easy to show that the reduction with respect to the subgroups shown in Figure 3 can be done as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} \bmod \begin{bmatrix} 1 & \\ & 4 \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix},$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \bmod \begin{bmatrix} 1 & \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 0 \\ (y - x) \bmod 4 \end{bmatrix}, \tag{23}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \bmod \begin{bmatrix} 4 & \\ & 1 \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}.$$

Note that the result of each reduction in (23) requires two bits to be encoded, and this is coherent with the fact that this scheme is 2-tight and non redundant.

Note that reductions (23) do not require arithmetic in a Galois field but only normal integer arithmetic. This can be interesting from an implementation point of view.

(2) If the two subgroups marked with full circles and the one marked with the star are chosen, one obtains a scheme that is not 2-tight anymore, since, for example, node 2 and node 6 have as common descendant node 13. The scheme, however, is not redundant, since every pair of nodes has as common ancestor node 1.

(3) If the two subgroups marked with full circles and the ones marked with a triangle are chosen (nodes 6, 11, and 12) one obtains a redundant scheme, since nodes 6 and 12 have as common ancestor node 9.

Remark 5. It is worth observing that the scheme proposed here is *not to be confused* with lattice-based error correction codes proposed in the literature (see [17] for an introduction). Generally speaking, lattice-based error correction schemes exploits the "metric" properties of lattices that derive from the fact that a lattice is subset of $\mathbb{R}^N$. In our case, we use the lattice only as an abstract group and not as the subset of a metric space. This distinction can be made clearer by observing that if one in an error correction scheme replaces the lattice with another one, almost surely the properties of the error correction scheme will change; in our scheme, one can replace the lattice with any other isomorphic group, and the overall properties of the scheme *will not change.*

5. Conclusions

This paper proposed a general framework for reduction schemes based on group theory. The new framework, while containing the vector space approach as a special case, allows to design schemes that cannot be modeled as linear codes. The properties of the GBRS have been analyzed, and it has also been shown how a GBRS can be used to prevent stream poisoning and how GBRS can be converted into a secret-sharing scheme achieving perfect secrecy. Examples of group-based schemes that cannot be described in the vector space framework have also been shown.

Appendices

A. Basic Concepts of Group Theory

In this paper, we are going to use some basic results and concepts from group theory. In order to make this paper as self-contained as possible, we recall here the main concepts used in this paper and refer the reader to the literature for more details [11].

If G is a group and H is a subgroup of G, H is said to be a *normal subgroup* of G (and will write $H \lhd G$) if for every $c \in G$ and $h \in H$, it holds $c^{-1}hc \in H$. If H is a subgroup of G, one can define their quotient G/H as the set of the classes associated with the equivalence relation $a \equiv b (\bmod H) \Leftrightarrow b^{-1}a \in H$. If $H \lhd G$, one can give to G/H the structure of a group by defining the group operation, as usual, by $(uH)(vH) := (uv)H$ [11]. If H is a subgroup of G we will denote with $\pi_H : G \to G/H$ the natural map associated with G/H, that is, the map that associates with each $x \in G$ the coset $\pi_H(x) := xH$ of G/H to which x belongs.

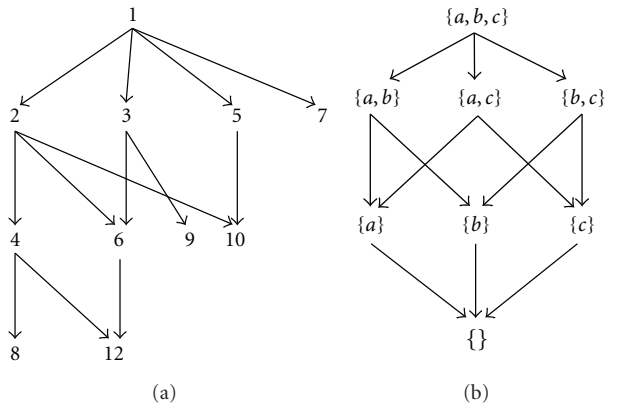

Figure 4: (a) The graph of the lattice of integers $\{1, \ldots, 12\}$ by divisibility. (b) The of the lattice of the subsets of $\{a, b, c\}$ ordered by inclusion.

If H and K are subgroups of G, we define

$$HK := \{hk,\ h \in H,\ k \in H\}. \tag{A.1}$$

It is known that if H and K are normal subgroups of G, then HK is a normal subgroup of G, and it is equal to KH [11].

B. Ordered Lattices

A structure that we will need in this paper is the *lattice* structure, a special type of partially ordered set.

Definition 7. A lattice L is a partially ordered set in which any two elements $a, b \in L$ have a least upper bound $a \vee b$ and a greatest lower bound $a \wedge b$ [11].

Many algebraic structures are lattices. The simplest example is maybe the set of subsets of a given set ordered by inclusion. In this case, the least upper bound corresponds to set union, while the least lower bound corresponds to set intersection. Another example of lattice is given by natural numbers ordered by divisibility (i.e., $a \geq b$ if a divides b). In this case, the least upper bound is the greatest common divisor, while the least lower bound corresponds is the least common multiple [11]. Here, we are interested in the lattice of *normal subgroups* of a given group. It is possible to show that the set of normal subgroups of G is a lattice, with set inclusion as the relation order, and that the least upper bound of H and K is HK, while the greatest lower bound is $H \cap K$.

B.1. Lattice Graph. Finite lattices have a useful graphic representation exploiting the idea of *covering*. We say that a *covers* b if "a is immediately above b;" that is, if $a > b$ and there exists no u such that $a > u > b$. We can represent the order relation by creating an oriented graph whose nodes are the lattice points, and there is an edge going from a to b if a covers b. Since it is possible to show that in a finite lattice $a > b$ if and only if there exist a sequence of c_i, $i = 1, \ldots, n$ such that (i) c_i covers c_{i+1} for every $i = 1, \ldots, n-1$ and (ii) $a = c_1$, $b = c_n$, it is easy to see that $a > b$ if and only if there is a path that goes from a to b. It is easy to verify that the least upper bound of a and b is the first common ancestor of a and b, while the greatest lower bound is the first common descendant.

Figure 4 shows two examples of lattice graphs. In Figure 4(a), one can see the graph of $\{1, \ldots, 12\}$ ordered by divisibility, while in Figure 4(b), one can see the graph of the subsets of $\{a, b, c\}$. It is common to draw the graph representing a lattice in order to have the edges always going from top to bottom.

References

[1] R. Bernardini, R. Rinaldo, and A. Vitali, "A reliable chunkless peer-to-peer architecture for multimedia streaming," in *Proceedings of the IEEE Data Compression Conference (DCC '08)*, pp. 242–251, Brandeis University, Snowbird, UT, USA, March 2008.

[2] S. Alstrup and T. Rauhe, "Introducing—a new technology for large-scale streaming over the internet," *EBU Technical Review*, no. 303, 2005.

[3] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: high-bandwidth multicast in cooperative environments," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, 2003.

[4] M. Yang and Y. Yang, "Peer-to-peer file sharing based on network coding," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS '08)*, pp. 168–175, Beijing, China, June 2008.

[5] R. Bernardini, R. C. Fabbro, and R. Rinaldo, "Peer-to-peer epi-transport protocol," 2011, http://tools.ietf.org/html/draft-bernardini-ppetp-02.

[6] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[7] P. Dhungel, X. Hei, K. W. Ross, and N. Saxena, "Pollution in P2P live video streaming," *International Journal of Computer Networks and Communications*, vol. 1, pp. 99–110, 2009.

[8] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, John Wiley and Sons, 1994.

[9] S. C. Kothari, "Generalized linear threshold scheme," in *Proceedings of the CRYPTO 84 on Advances Incryptology*, pp. 231–241, Springer, 1985.

[10] R. Bernardini, R. C. Fabbro, and R. Rinaldo, "Peer-to-peer streaming based on network coding improves packet jitter," in *Proceedings of the ACM International Conference on Multimedia*, Florence, Italy, October 2010.

[11] N. Jacobson, *Basic Algebra. I*, Freeman, New York, NY, USA, 1985.

[12] W. Scott, *Group Theory*, Dover, New York, NY, USA, 1987.

[13] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[14] T. M. Cover and J. A. Thomas, *Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.

[15] V. Kucera, *Discrete Linear Control: The Polynomial Equation Approach*, John Wiley & Sons, New York, NY, USA, 1979.

[16] R. Bernardini, "An efficient algorithm to find lattice chains," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 5, pp. 496–505, 1996.

[17] J. H. Conway, N. J. A. Sloane, and E. Bannai, *Sphere-Packings, Lattices, and Groups*, Springer, New York, NY, USA, 1987.