Distributed Multiparty DC Power Flow Algorithm with Secure Exchange of Information

Sanja Cvijić and Marija Ilić

Electrical and Computer Engineering, Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213, USA
sanja13@andrew.cmu.edu, milic@ece.cmu.edu
http://www.ece.cmu.edu/

Abstract. This paper introduces a new distributed algorithm for computing DC power flow with secure information exchange among participating areas. The algorithm is based on the principles of the Diakoptics approach to distributed power flow execution. Power flow is computed in multiple steps through a precisely defined information exchange protocol between the coordinator and the individual areas. In the conventional Diakoptics algorithm, areas exchange the information about their generation and load with the coordinator. Our algorithm redefines the exchange of information protocol so that areas do not need to reveal their internal generation and load but only their effects on the rest of the grid. In this way, proprietary information about the exact locations and values of generation and demand are protected and cannot be uniquely deciphered. We illustrate this algorithm and the corresponding information exchange protocol on the IEEE 14-bus system.

Keywords: cyber security, distributed power flow, diakoptics, information exchange, communication.

1 Introduction

The current rate of increase in power grid size and complexity requires improved distributed power flow algorithms. Additionally, interconnected areas in a deregulated environment are strongly motivated to increase self-sufficiency and reveal the minimum amount of information necessary for reliable system operation. Consequently, distributed algorithms designed for power systems should fit the needs of competing areas while taking into consideration the sensitive nature of exchanged information.

The "DC" simplification of the full AC power flow problem is widely used in power systems analysis for market related purposes. Using "DC" assumptions, areas can asses the effect of energy transactions across areas on active power flows in order to prevent transmission line overloading. Additionally, "DC" power flow is widely used in contingency analysis for assessing active power line flows.

A distributed power flow implementation assumes that computations are divided across multiple parties who jointly solve a system-level problem through

D.C. Tarraf (ed.), *Control of Cyber-Physical Systems*,
Lecture Notes in Control and Information Sciences 449,
DOI: 10.1007/978-3-319-01159-2_13,

communication. Although the distributed execution brings high performance benefits, it also puts the grid at risk if the flow of information is intercepted. Recent trends in power grids show that the overall grid vulnerability continues to increase with emerging communication and networking protocols. However, the majority of distributed algorithms do not address the question of the security of the exchanged data nor the willingness of parties to exchange sensitive information. For economic reasons in competitive electricity markets, the question about willingness of areas to share sensitive information plays an important role. Areas tend to keep the outputs and locations of their generators private in order to protect their bidding strategies.

Various cryptographic approaches have been proposed for maintaining the privacy of sensitive data in power systems. The best-known one is the Secure Multiparty Computation (SMC) algorithm used to carry out extensive computations of a function among multiple parties [1],[2]. The SMC problem assumes that multiple parties can jointly and accurately compute a function over their inputs while keeping them private. Clearly, a similar concept is needed for computing power flow securely. Since the goal is to protect information about their own generation and load, areas in an interconnected multi-area electric system need to solve power flow jointly without revealing their inputs. The main disadvantage of general cryptographic methods is that they increase computational overhead in order to achieve privacy.

In the literature, several methods have been proposed for solving distributed power flow. In the Sixties, Diakoptics was introduced by G. Kron [3], and applied to power systems by H. Happ [4],[5]. Although, the original formulation of this algorithm was in terms of electric circuit variables, in this paper, we extend it to the "DC" power flow problem. The "DC" (or simply DC) acronym denotes an approximation of the full AC power flow problem. When analogies between electrical and DC power networks [6] are applied, it becomes clear that the original Diakoptics algorithm represents one of the first distributed power flow algorithms. However, this algorithm requires areas to communicate their internal generation and demand with the coordinator making the grid vulnerable to cyber attacks.

Recently, new contributions have been made in computing power flow in a fully decentralized way through communication with nearest neighbors only. Distributed AC power flow can be solved by installing small line and bus calculators that, through the exchange of information with nearest neighbors, can compute power flow in a distributed way [7], [8]. Regarding the communication aspect, this algorithm requires lines to exchange flow variables with adjacent buses and buses to exchange Lagrange multipliers with directly connected lines. Due to Kirchhoff's Flow Law, it should be noted that line flow variables uniquely reveal power injections into buses because all together sum to zero.

A distributed power flow algorithm based on determining the Thevinan equivalent for the exterior network and gradually correcting the equivalent power injection of the exterior system has been presented in [9]. This method is based on equivalencing which does not require areas to share their nodal injections with

the coordinator. Due to the nature of equivalencing, topological and operational data are merged together requiring all computations to be repeated as network conditions change. Inability to re-use topological information, in a network in which only loading conditions change, increases the overall computational complexity of the algorithm.

As an alternative approach, we are proposing a new non-cryptographic method for solving distributed power flow in a secure manner. The idea is to find a solution to the power flow problem in a distributed way while keeping nodal injections private. The power flow problem cannot be solved in totally isolated areas with no information exchange since areas are electrically coupled. This is why some communication is required for solving power flow accurately. Computation of the effect of interconnections generally requires the exact values of generation and demand to be shared. In our proposed algorithm, the effect of interconnections can be accurately incorporated into the power flow solution by each area without explicit knowledge of nodal injections. The main idea is that areas compute how their internal injections affect the external network while the coordinator computes how the flows in the external interconnections affect power flow inside areas.

2 Algorithm for Secure Distributed Power Flow

This multi-layered algorithm originates from the Diakoptics approach explained in detail by H. Happ in [5]. Inspired by this procedure which was initially designed for electric circuits, we have formulated a coordinated distributed DC power flow algorithm and have applied it to more efficient contingency screening in [10]. For the sake of comparison, this algorithm is revisited in this paper and referred to as the Simple Distributed DC Power Flow Algorithm. As part of that algorithm, it was necessary that individual areas exchange their generation and load data with the coordinator. In this paper, areas send coded information about the effect of their generation and load onto the rest of the network. The coded information is insufficient for uniquely restoring areas' generation and demand, but sufficient for solving the multi-area power flow problem.

The goal of Diakoptics is to enable distributed execution through decomposition into two layers: one that operates on disconnected areas while neglecting interconnections with the rest of the system and another that integrates the effects of interconnections. These layers emphasize the modeling level of interest and will be referred to as *zoomed-in* and *zoomed-out* layers respectively.

At the zoomed-in layer, power flow is computed within individual areas completely disconnected from the rest of the system. Due to Kirchhoff's laws and Ohm's law, there exist couplings among all areas in the system, i.e., power generation in one area affects line flows in another. The zoomed-out layer, or the coordinating layer, quantifies the effect of interconnections by operating on a simplified representation of the original network. Finally, each area updates its power flow solution by incorporating that effect into the solution previously computed when the area was disconnected.

2.1 Mesh-to-Tree Transformation

The key idea that enables distributed execution of power flow lies in decomposing and simplifying a multi-area network. The network topology is broken down into subdivisions, or areas, and then simplified through transformation of a meshed electrical network into a spanning tree [11]. This graph-based transformation is first applied at the zoomed-in layer so that topologies of individual areas are converted into trees in parallel. Then, spanning trees are interconnected with tie-lines and transformations proceed at the zoomed-out layer until the initial network has been converted into a tree.

The motivation for converting a meshed network into a tree comes from the fact that line flows in a spanning tree have a physical interpretation. They can be understood as bilateral transactions or financial contracts that specify the amount of traded power along a contract path that the trade is supposed to take. However, physical laws that govern power grids disrupt the intended path by superposing loop flows inside basic loops in a meshed network. Additionally, due to Kirchhoff's Voltage Law, only line voltages across spanning tree branches can be considered independent while all remaining voltages can be expressed in terms of tree voltages. Therefore, the spanning tree contains the minimum information about the network. A spanning tree can be considered a financial network with trades assigned along its branches.

All network transformations obey the law of conservation of power while each network representation satisfies Kirchhoff's Flow Law (KFL), Kirchhoff's Voltage Law (KVL) and Ohm's law. A network representation, also called a *reference frame*, is fully described with a flow vector F, a voltage angle difference vector Δ and a reactance matrix X. Mapping between two reference frames (a and b) is established through a connection matrix C_{ab}. Assuming that variables in one reference frame are known, it is possible to transform them into the corresponding variables in another reference frame. Starting from the physical law equations, the resulting relationships follow:

$$F^a = C_{ab} \cdot F^b \tag{1}$$

$$\Delta_b = C_{ab}^T \cdot \Delta_a \tag{2}$$

$$X_{bb} = C_{ab}^T \cdot X_{aa} \cdot C_{ab} \tag{3}$$

The algorithm consists of five reference frames in which computations are performed in two directions, as shown in the flowchart in Figure 1. Topological transformations are performed in the *forward* direction from reference frame 1 to 5, while power flow is solved in the *backward* direction from reference frame 5 to 1.

We briefly illustrate the method on the 14-bus system in Figure 2, which has been clustered into three areas using spectral clustering with electrical distances as the measure of distance [12].

Zoomed-In Layer: After decomposing a large network into areas, the zoomed-in layer analyzes each area as if the others did not exist. Initially, it performs

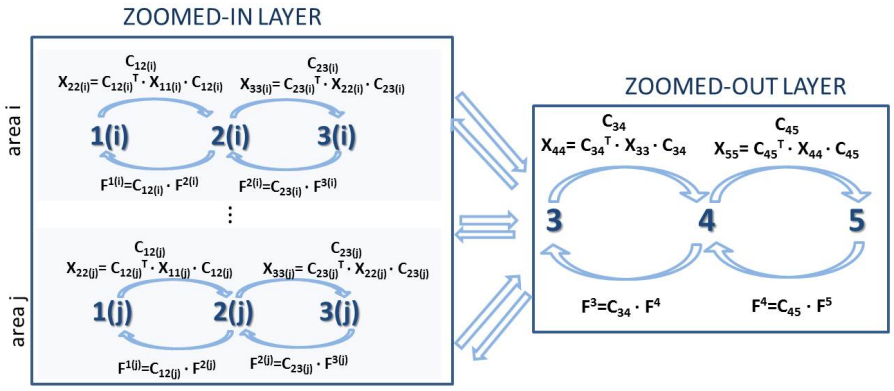


Fig. 1. Algorithm flowchart

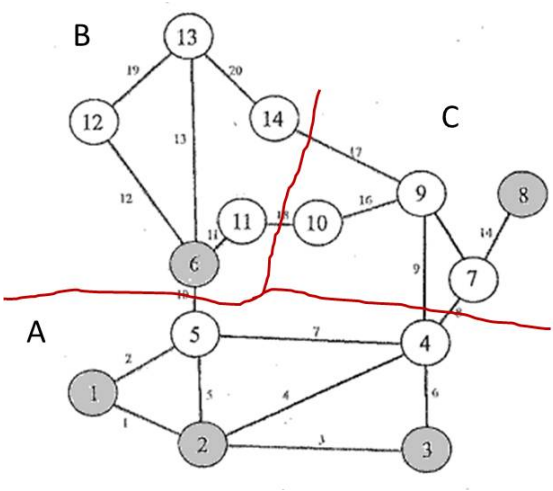


Fig. 2. Clustered 14-bus system

transformations of internal area topologies into spanning trees in a distributed way. It consists of three reference frames that gradually lead to the spanning tree representation. *Reference frame 1* represents the original meshed network which is mapped into *reference frame 2* which is composed of spanning tree branches and basic loops. The choice of a spanning tree is arbitrary; however, it uniquely determines a set of basic loops. It can be shown that, due to KVL, basic loops can be eliminated, resulting in a completely radial network representation. This representation is referred to as *reference frame 3*.

The mesh-to-tree transformation is illustrated on area A of the 14-bus example. On the zoomed-in layer, network topologies of all three areas are converted into spanning tree representations. The connection between reference frames 1 and 2

is established through a connection matrix, C_{12}, as shown in Equation (4). The C_{12} matrix defines mapping between flows in reference frames 1 and 2, depicted in Figure 3. It is determined by inspection, through comparison of corresponding flows. For example, flow f^1 through line 1-2 is a summation of tree flow $f^{1'}$ and loop flow $-f^{c1}$, where the negative sign in front of f^{c1} is due to the mismatch in the direction of the loop flow and the f^1 flow in the physical network.

$$
\begin{aligned}
F^{1(A)} &= C_{12(A)} \cdot F^{2(A)} \\
\begin{bmatrix} f^1 \\ f^2 \\ f^3 \\ f^4 \\ f^5 \\ f^6 \\ f^7 \end{bmatrix} &= \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right] \cdot \begin{bmatrix} f^{1'} \\ f^{2'} \\ f^{3'} \\ f^{4'} \\ \hline f^{c1} \\ f^{c2} \\ f^{c3} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} F^{prime} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} F^{ic} \end{matrix}
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
\Delta_{2(A)} &= X_{22(A)} \cdot F^{2(A)} \\
\left[\frac{\Delta_{prime}}{0}\right] &= \left[\frac{X_{prime\_prime} \mid X_{prime\_ic}}{X_{ic\_prime} \mid X_{ic\_ic}}\right] \cdot \left[\frac{F^{prime}}{F^{ic}}\right]
\end{aligned}
\tag{5}
$$

In reference frame 2, the sum of phase angle differences along any loop in an electric power network is equal to zero due to KVL, Equation (5). Consequently, line flows in reference frame 2 are linearly dependent. Moreover, loop flows can be expressed as a linear combination of spanning tree flows as in Equation (6).

$$
\begin{aligned}
F^{ic} &= -X_{ic\_ic}^{-1} \cdot X_{ic\_prime} \cdot F^{prime} = C_{ic\_prime} \cdot F^{prime} \\
&= \sum_{i=1}^{N} c_i \cdot F^{prime}
\end{aligned}
\tag{6}
$$

Since loop flows are dependent on spanning tree flows, they can be eliminated while the mapping between them can be saved in form of a connection matrix, C_{23}, Equation (7). The final area representation contains spanning tree flows only, F^{prime} (called F^3 from here on), depicted in Figure 4.

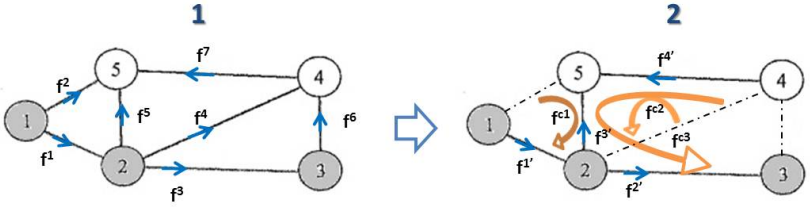


Fig. 3. Reference frame transition from 1 to 2 in area A

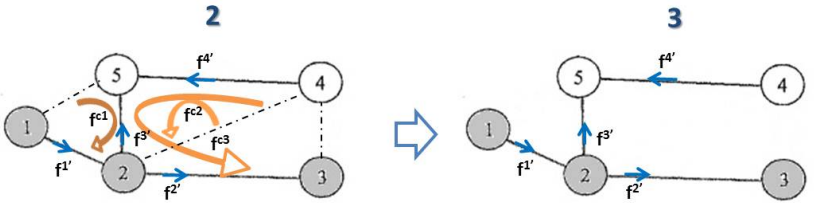


Fig. 4. Reference frame transition from 2 to 3 in area A

$$F^{2(A)} = C_{23(A)} \cdot F^{3(A)}$$

$$\begin{bmatrix} f^{1'} \\ f^{2'} \\ f^{3'} \\ f^{4'} \\ \hline f^{c1} \\ f^{c2} \\ f^{c3} \end{bmatrix} = \begin{bmatrix} 1_{4\times 4} \\ \hline C_{ic\_prime} \end{bmatrix} \cdot \begin{bmatrix} f^{1'} \\ f^{2'} \\ f^{3'} \\ f^{4'} \end{bmatrix} \tag{7}$$

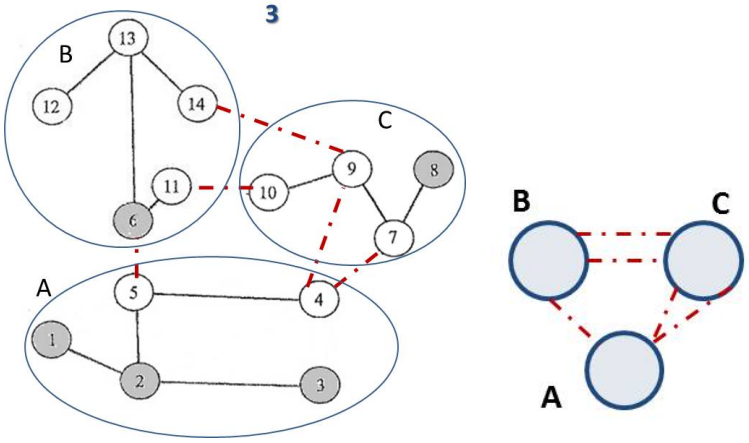
At this point, areas send their spanning tree representations and nodal injections to the coordinator who computes the effect of interconnections. Afterwards, areas receive their updated spanning tree flows, denoted as F^3, which contain flows due to internal injections, F^{ist}, as well as flows due to couplings with the rest of the network, EFF:

$$F^3 = F^{ist} + EFF \tag{8}$$

Once the F^3 flows are known, spanning tree flows can be mapped into physical line flows through reference frame 2 as in Equation (9).

$$\begin{matrix} F^2 = C_{23} \cdot F^3 \\ F^1 = C_{12} \cdot F^2 \end{matrix} \quad or \quad \begin{matrix} C_{13} = C_{12} \cdot C_{23} \\ F^1 = C_{13} \cdot F^3 \end{matrix} \tag{9}$$

Zoomed-Out Layer: The zoomed-out layer refers to network transformations which compute the effect of interconnections necessary for solving power flow accurately. First, tree representations of all areas are interconnected with tie-lines, as in Figure 5. The connection and reactance matrices that describe this interconnected network representation are created by stacking connection, $C_{13(i)}$, and reactance, $X_{33(i)}$, matrices of the zoomed-in layer and tie-lines on the main diagonal. The goal of the zoomed-out layer is to neglect the exact network topology inside an area while merging it into the slack bus. This is accomplished by



(a) Interconnected spanning trees (b) Zoomed-out view

Fig. 5. Zoomed-out representation of 14-bus system

keeping exactly one tie-line connection between slack buses of each two interconnected areas. Remaining tie-line flows are substituted with external loop flows between two areas, shown in Figure 6. This zoomed-out network representation is called *reference frame 4*. After zooming-out, the same graph-based transformations, as were used on the zoomed-in layer, are applicable for elimination of basic loops. Spanning tree branches and basic loops are identified to form a new network representation, *reference frame 5*. Starting from Ohm's law and KVL in Equation (10), external loop flows in the zoomed-out representation, F^{ec}, can be expressed as contributions of internal, F^{ist}, and external, F^{est}, spanning tree flows, Equation (11).

$$\begin{aligned} \Delta_5 &= X_{55} \cdot F^5 \\ \left[\begin{array}{c} \Delta_{ist} \\ \hline \Delta_{est} \\ \hline 0 \end{array}\right] &= \left[\begin{array}{c|c|c} X_{ist\_ist} & X_{ist\_est} & X_{ist\_ec} \\ \hline X_{est\_ist} & X_{est\_est} & X_{est\_ec} \\ \hline X_{ec\_ist} & X_{ec\_est} & X_{ec\_ec} \end{array}\right] \cdot \left[\begin{array}{c} F^{ist} \\ \hline F^{est} \\ \hline F^{ec} \end{array}\right] \end{aligned} \tag{10}$$

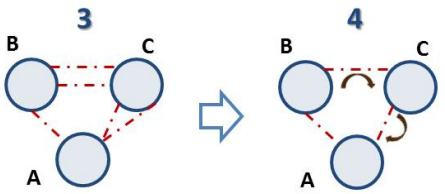


Fig. 6. Reference frame transition from 3 to 4 on the zoomed-out layer

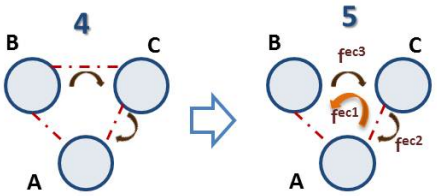


Fig. 7. Reference frame transition from 4 to 5 on the zoomed-out layer

$$\begin{aligned} F^{ec} &= -X_{ec\_ec}^{-1} \cdot \left(X_{ec\_ist} \cdot F^{ist} + X_{ec\_est} \cdot F^{est} \right) \\ &= C_{ec\_ist} \cdot F^{ist} + C_{ec\_est} \cdot F^{est} \end{aligned} \tag{11}$$

Reference frame 5 is the final representation that captures internal and external spanning tree flows along with their contributions to external loop flows. Internal tree flows, F^{ist}, originate from internal nodal injections. External tree flows, F^{est}, contain information about injections into slack buses of individual areas. With reference frame 5, the topological transformations have been completed.

2.2 Simple Distributed DC Power Flow Algorithm

Once all transformations have been performed in the forward direction, power flow computation can take place in the backward direction, Figure 1. The computation of power flow starts with reference frame 5 in which all internal and external spanning tree flows have to be known. Conventionally, inputs into the power flow problem are nodal power injections. It can be shown that for a selected spanning tree, there exists a bijective mapping between specified nodal injections and spanning tree flows. Here we prove the bijective mapping in the form of two theorems.

Theorem 1: *Nodal injections uniquely determine line flows in any radial network in which KFL holds.*

In a general network with N nodes, one node, known as slack, is responsible for total power balancing while nodal injections are specified at the remaining $N-1$ nodes. The goal is to find $N-1$ spanning tree flows, F^{ist}, created by nodal injections so that KFL is satisfied. By setting up KFL balance equations for each node that is not slack, we formulate a system of $N-1$ linearly independent equations with $N-1$ unknown flow variables. This relationship can be expressed in the form of matrix T which contains only ± 1 and 0 elements, Equation (12). Since matrix T is nonsingular, there exists a unique set of spanning tree flows for the specified nodal injections, Equation (13).

$$P^{inj} = T \cdot F^{ist} \tag{12}$$

$$F^{ist} = T^{-1} \cdot P^{inj} \tag{13}$$

For example in a five-bus network with bus 1 as slack, nodal injections are specified on the left while the corresponding spanning tree flows are shown on the right in Figure 8. The corresponding T matrix that translates tree flows into injections (left) and the T^{-1} matrix that translates injections into tree flows (right) are shown in Equation (14).

$$\begin{bmatrix} p^2 \\ p^3 \\ p^4 \\ p^5 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} f^{A1} \\ f^{A2} \\ f^{A3} \\ f^{A4} \end{bmatrix} \quad or \quad \begin{bmatrix} f^{A1} \\ f^{A2} \\ f^{A3} \\ f^{A4} \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} p^2 \\ p^3 \\ p^4 \\ p^5 \end{bmatrix} \tag{14}$$

Theorem 2: *Line flows uniquely determine nodal injections in any radial network in which KFL holds.*

The proof of this theorem directly follows from Equation (12).

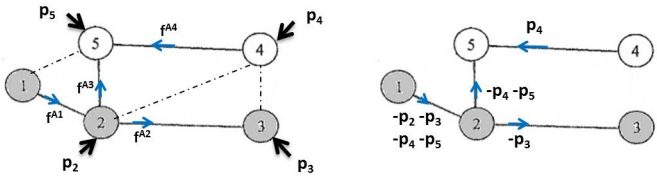


Fig. 8. Left: Nodal injections, Right: Spanning tree flows

Once nodal injections are converted into internal and external spanning tree flows using Theorem 1, power flow can be solved. Firstly, on the zoomed-out layer, external loop flows are computed from spanning tree flows using Equation (11). Secondly, flows from reference frame 5 are translated across reference frame 4 to reference frame 3 using connection matrices.

$$\begin{matrix} F^4 = C_{45} \cdot F^5 \\ F^3 = C_{34} \cdot F^4 \end{matrix} \quad or \quad \begin{matrix} C_{35} = C_{34} \cdot C_{45} \\ F^3 = C_{35} \cdot F^5 \end{matrix} \tag{15}$$

Further on, zoomed-in areas acquire updated spanning tree flows, F^3, computed on the zoomed-out layer. Finally, each area i maps these tree flows into physical line flows using its own connection matrix:

$$F^{1(i)} = C_{13(i)} \cdot F^{3(i)} \tag{16}$$

A diagram that summarizes the simple distributed power flow algorithm with the exchange of information is shown in Figure 9. In order to allow the mesh-to-tree transformation, each area needs to send their tree representation with the corresponding tree reactance matrix, $X_{33(i)}$, and all internal spanning tree flows, $F^{ist(i)}$, translated from its nodal injections. After the zoomed-out transformations, the coordinator sends updated spanning tree flows, $F^{3(i)}$, to the individual areas that then use them for computing physical line flows, $F^{1(i)}$.

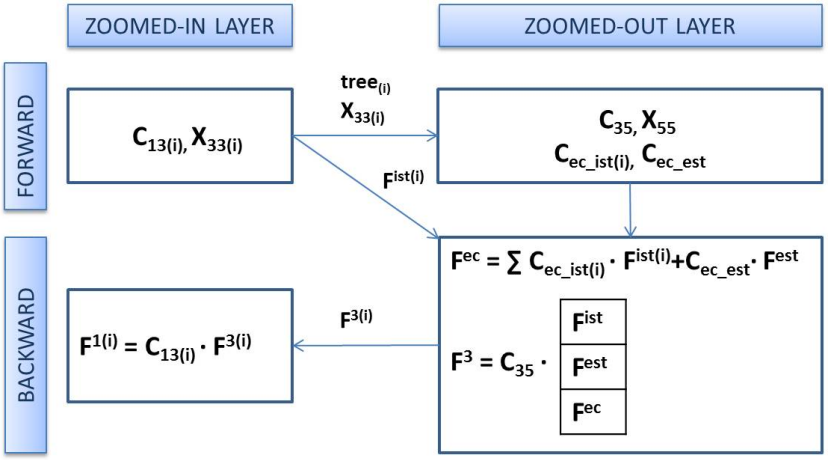


Fig. 9. Exchange of information in the simple DC power flow algorithm

2.3 Secure Distributed DC Power Flow Algorithm

In the simple distributed DC power flow formulation, all areas are required to exchange their internal spanning tree flows with the coordinator. From this exchange of information, location and outputs of each generator/load can be easily decoded based on Theorem 2. In order to overcome these security issues, we propose a new algorithm for distributed power flow execution. Compared to the previously introduced algorithm, the secure algorithm does not require areas to reveal their generation and load locations. Even if the entire flow of information has been intercepted, it is not possible to decode the locations and outputs of injection patterns of individual areas.

The key idea of this approach is based on the exchange of mutual effects between zoomed-in and zoomed-out layers. Loop flows create couplings among areas and therefore couple zoomed-in and zoomed-out layers. Areas can exchange their effects on external loop flows with the zoomed-out layer, instead of revealing their internal generation and demand. Similarly, the zoomed-out layer exchanges effects of external spanning tree flows and external loop flows on spanning tree flows inside areas, so that areas can calculate their own physical line flows. We show that it is not possible to decode information about neither internal spanning tree flows nor external spanning tree flows. This method provides an attack-resistant flow of information which successfully hides secure information.

Instead of exchanging information two-ways as in the simple distributed DC algorithm, this method requires a four-ways exchange of information, Figure 10. Starting with the zoomed-in layer, areas convert their topologies into trees and forward their tree representations to the zoomed-out layer. Further on, the zoomed-out layer interconnects spanning trees and proceeds with tree transformations in order to compute matrices C_{ec_ist} and C_{ec_est}, Equation (11). These matrices map internal and external spanning tree flows into external loop flows.

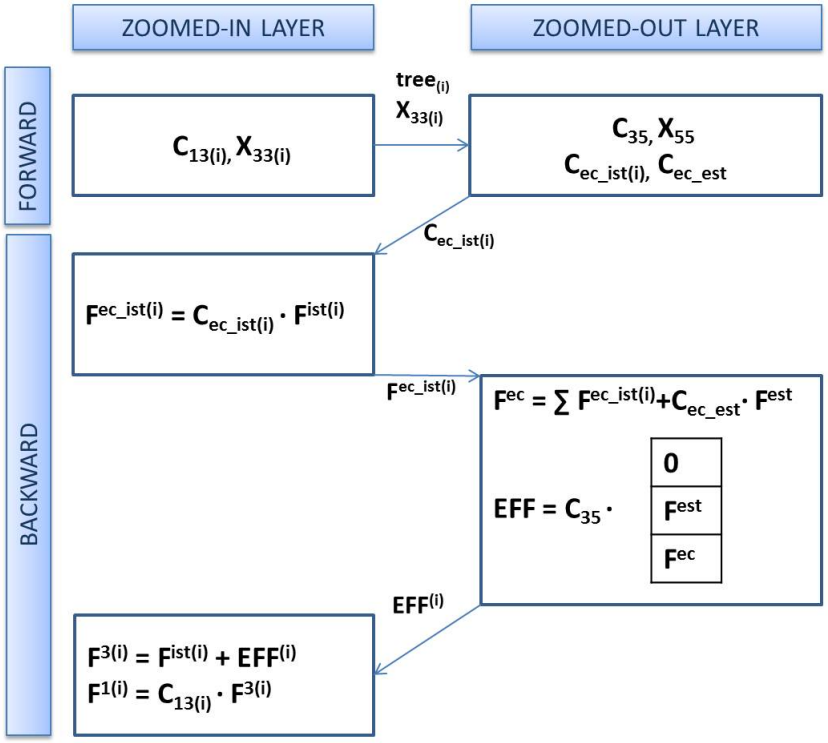


Fig. 10. Exchange of information in the secure DC power flow algorithm

The newly computed C_{ec_ist} matrix is divided into blocks corresponding to individual areas. Area i computes the effect of its internal tree flows onto external loop flows, $F^{ec\_ist(i)}$:

$$F^{ec\_ist(i)} = C_{ec\_ist(i)} \cdot F^{ist(i)} \tag{17}$$

At this point, the coordinator knows how areas affect external loop flows which enables the exact computation of external loop flows. The coordinator then sums up the effects of all areas onto external loop flows and adds them to the loop flows created by external spanning tree flows:

$$F^{ec} = \sum_i F^{ec\_ist(i)} + C_{ec\_est} \cdot F^{est} \tag{18}$$

Now, it is possible to determine the exact effect of interconnections onto individual areas. This effect is computed on the zoomed-out layer assuming that all internal spanning tree flows equal zero:

$$EFF = C_{35} \cdot \begin{bmatrix} 0 \\ \hline F^{est} \\ \hline F^{ec} \end{bmatrix} \tag{19}$$

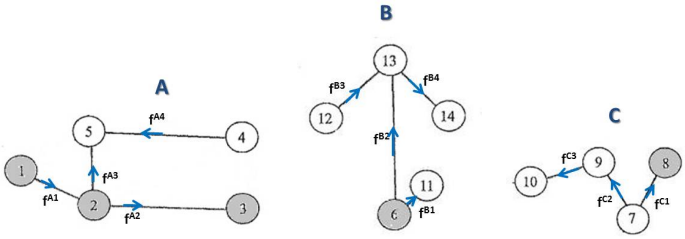
Individual areas receive the effect of interconnections on their tree flows which is added to their internal spanning tree flows. Then, the tree flows are mapped back to physical line flows using connection matrix $C_{13(i)}$.

3 Simulation Results

The secure distributed DC power flow algorithm is illustrated on the IEEE 14-bus example, Figure 2. Line parameters and generation and load magnitudes are given in Appendix A. Internal and external spanning tree flows are created from specified nodal injections (Table 2) based on Theorem 1, and given in Equation (20). After performing the mesh-to-tree transformation, tree representations of areas are shown in Figure 11 and the corresponding connection and reactance matrices are given in Equations (21) and (22) respectively.

$$F^{ist(A)} = \begin{bmatrix} 1.3130 \\ 0.9420 \\ 0.5540 \\ -0.4780 \end{bmatrix}, F^{ist(B)} = \begin{bmatrix} 0.0350 \\ 0.3450 \\ -0.0610 \\ 0.1490 \end{bmatrix}, F^{ist(C)} = \begin{bmatrix} -0.3850 \\ 0.3850 \\ 0.0900 \end{bmatrix}$$
$$F^{est} = \begin{bmatrix} 0.4920 \\ 0.3850 \end{bmatrix} \tag{20}$$

$$C_{13(A)} = \begin{bmatrix} 0.8383 & -0.0907 & -0.2288 & -0.0597 \\ 0.1617 & 0.0907 & 0.2288 & 0.0597 \\ -0.0271 & 0.5600 & 0.1293 & -0.0506 \\ -0.0567 & 0.2020 & 0.2706 & -0.1059 \\ -0.0778 & 0.1472 & 0.3713 & 0.0969 \\ -0.0271 & -0.4400 & 0.1293 & -0.0506 \\ -0.0839 & -0.2379 & 0.3999 & 0.8435 \end{bmatrix}$$
$$C_{13(B)} = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 0.2223 & -0.3411 & 0 \\ 0 & 0.7777 & 0.3411 & 0 \\ 0 & 0.2223 & 0.6589 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$
$$C_{13(C)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$



(a) Area A (b) Area B (c) Area C

Fig. 11. Spanning trees of areas on the zoomed-in layer

$$X_{33(A)} = \begin{bmatrix} 0.0496 & -0.0054 & -0.0135 & -0.0035 \\ -0.0054 & 0.1109 & 0.0256 & -0.0100 \\ -0.0135 & 0.0256 & 0.0646 & 0.0168 \\ -0.0035 & -0.0100 & 0.0168 & 0.0355 \end{bmatrix}$$

$$X_{33(B)} = \begin{bmatrix} 0.1989 & 0 & 0 & 0 \\ 0 & 0.1013 & 0.0444 & 0 \\ 0 & 0.0444 & 0.1317 & 0 \\ 0 & 0 & 0 & 0.3480 \end{bmatrix}$$

$$X_{33(C)} = \begin{bmatrix} 0.1762 & 0 & 0 \\ 0 & 0.1100 & 0 \\ 0 & 0 & 0.0845 \end{bmatrix} \tag{22}$$

The zoomed-out layer calculates contribution coefficients of internal spanning tree flows of individual areas and contribution coefficients of external spanning tree flows to external loop flows according to Equation (11):

$$C_{ec\_ist(A)} = \begin{bmatrix} -0.0045 & -0.0129 & 0.0217 & 0.0457 \\ -0.0017 & -0.0047 & 0.0079 & 0.0167 \\ -0.0027 & -0.0078 & 0.0131 & 0.0275 \end{bmatrix}$$

$$C_{ec\_ist(B)} = \begin{bmatrix} 0.1542 & 0.0519 & 0.0228 & 0.1783 \\ 0.0562 & 0.0189 & 0.0083 & 0.0650 \\ 0.2593 & -0.0535 & -0.0235 & -0.1838 \end{bmatrix}$$

$$C_{ec\_ist(C)} = \begin{bmatrix} 0 & -0.0900 & -0.0655 \\ 0 & 0.0929 & -0.0239 \\ 0 & -0.0542 & -0.1102 \end{bmatrix}$$

$$C_{ec\_est} = \begin{bmatrix} 0.3416 & -0.1997 \\ 0.1246 & 0.1661 \\ 0.2057 & -0.1202 \end{bmatrix} \tag{23}$$

Based on the tracing coefficients, individual areas can compute the effect of their internal spanning tree flows to external loop flows using Equation (17). External loop flows created by internal tree flows have magnitudes as follows

	$F^{ec\_ist(A)}$	$F^{ec\_ist(B)}$	$F^{ec\_ist(C)}$
f^{ec1}	−0.0280	0.0485	−0.0405
f^{ec2}	−0.0102	0.0177	0.0336
f^{ec3}	−0.0168	−0.0353	−0.0308

(24)

Once the zoomed-out layer collects all external loop flows created by individual areas, they are summed up together with loop flows created by external tree flows according to Equation (18). The magnitude of external loop flows are

	F^{ec}
f^{ec1}	0.0712
f^{ec2}	0.1663
f^{ec3}	−0.0280

(25)

Starting from the relationship in Equation (19), the zoomed-out layer can compute the effect of interconnections on areas' spanning tree flows:

$$EFF^{(A)} = \begin{bmatrix} 0.8770 \\ 0 \\ 0.8770 \\ -0.4562 \end{bmatrix}, EFF^{(B)} = \begin{bmatrix} 0.0280 \\ -0.0992 \\ 0 \\ -0.0992 \end{bmatrix}, EFF^{(C)} = \begin{bmatrix} 0.3850 \\ -0.0951 \\ -0.0280 \end{bmatrix} \tag{26}$$

The coordinator computes physical tie-line flows as well:

$$EFF^{tie\_lines} = \begin{bmatrix} 0.2899 \\ 0.1663 \\ 0.4208 \\ 0.0992 \\ -0.0280 \end{bmatrix} \tag{27}$$

Once areas integrate the effect of interconnections onto their internal spanning tree flows, physical line flows can be accurately computed:

$$F^{1(A)} = \begin{bmatrix} 1.4788 \\ 0.7112 \\ 0.7005 \\ 0.5523 \\ 0.4090 \\ -0.2415 \\ -0.6234 \end{bmatrix}, F^{1(B)} = \begin{bmatrix} 0.0630 \\ 0.0755 \\ 0.1703 \\ 0.0145 \\ 0.0498 \end{bmatrix}, F^{1(C)} = \begin{bmatrix} 0 \\ 0.2899 \\ 0.0620 \end{bmatrix} \tag{28}$$

4 Security Considerations

In the previous section, a new algorithm for secure distributed computation of DC power flow was introduced. The main advantage of this algorithm versus the simple distributed power flow algorithm is that it keeps internal spanning tree flows of areas private. Internal spanning tree flows uniquely map into nodal power injections using Theorem 2. Therefore, nodal power injections are kept private and it is not possible to recover them from the information exchange protocol.

The only exchanged information in the secure algorithm that incorporates internal power injections are effects of internal spanning tree flows to external loop flows, $F^{ec\_ist(i)}$. An additional information exchanged are contribution factors of internal spanning tree flows to external loop flows, $C_{ec\_ist(i)}$. The relationship between them can be stated as in Equation (17). However, the $C_{ec\_ist(i)}$ matrix is generally singular with dimensions: the number of external loop flows by the

number of nodes in area i minus one. Due to its singular nature, this matrix cannot be inverted in order to decipher spanning tree flows, $F^{ist(i)}$.

5 Possible Extensions

The distributed method for solving DC power flow can be generalized into a distributed DC Optimal Power Flow (DCOPF) algorithm with secure information exchange. In paper [13], we introduced a generalized DCOPF formulation that includes wheeling loop flow charges. The key idea is that each area determines their optimal dispatch so that wheeling across third-party areas is reduced. This algorithm requires that an area exchanges only information about external loop flows created by its own dispatch. In this paper, it was shown that the loop flow information does not uniquely reveal nodal injections and therefore it can be considered secure.

Another possible extension of the algorithm is for distributed AC power flow with secure information exchange. The presented algorithm strongly relies on the linear assumption inherent in DC power flow and therefore a further extension into the non-linear AC power flow requires major breakthroughs. One potential approach, which preserves the assumption about linearity, would use the constant impedance load model. With loads modeled as constant impedance, the AC problem is linear which would make our framework fully applicable. If the load P,Q model is preserved, thesis [14] defines another approximation of the full AC load flow problem around the base case loading conditions. This approximation relies on the derivation of coefficients which relate power nodal injections to power loop flows in a linear fashion.

6 Conclusions

This paper has introduced a new method for solving distributed DC power flow with secure information exchange. We have shown how it is possible to protect information about generation and load locations and power outputs while obtaining the correct power flow solution. Although the method is non-cryptographic, the decentralized nature of the algorithm successfully hides the sensitive information. Delegating power flow computation to areas and the coordinator according to the type of information that they have access to, protects the delicate information but also allows solving power flow accurately through the exchange of intermediate results. Compared to the simple distributed power flow algorithm, the secure algorithm requires more communication for the sake of improving privacy of the data.

Acknowledgment. This work was done as a part of the 2111.004 task of the SRC ERI research initiative http://www.src.org/program/eri/. The authors would like to thank SRC Smart Grid Research Center and its member companies on their financial support which made this work possible.

References

1. Cramer, R., Damgard, I., Nielsen, J.: Multiparty Computation, an Introduction. Contemporary Cryptology, 41–87 (2005)
2. Yao, A.C.: Protocols for secure computations. In: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, SFCS 1982, pp. 160–164 (1982)
3. Kron, G.: Diakoptics: The Piecewise Solution of Large-Scale Systems. Macdonald (1963)
4. Happ, H.H.: Diakoptics and Piecewise Methods. IEEE Transactions on Power Apparatus and Systems 89 (1970)
5. Happ, H.H.: Piecewise Methods and Applications to Power Systems. John Wiley & Sons, Inc., New York (1980)
6. Ilić, M., Zaborszky, J.: Dynamics and Control of Large Electric Power Systems. John Wiley & Sons (2000)
7. Ilić, M., Hsu, A.: Toward Distributed Contingency Screening Using Line Flow Calculators and Dynamic Line Rating Units (DLRs). IEEE Transactions on Sustainable Energy 2, 37–49 (2011)
8. Hsu, A., Ilić, M.: Distributed Newton Method For Computing RealDecoupled Power Flow In Lossy Electric Energy Networks. In: North American Power Symposium (NAPS) (2012)
9. Haibo, Z., Boming, Z., Hongbin, S., Ran, A.: A new distributed power flow algorithm between multi-control-centers based on asynchronous iteration. In: International Conference on Power System Technology (2006)
10. Cvijić, S., Ilić, M.: Contingency Screening in Multi-Control Area System Using Coordinated DC Power Flow. In: ISGT Europe Manchester, UK (2011)
11. Cvijić, S., Ilić, M.: On Limits to the Graph-Theoretic Approaches in the Electric Power Systems. In: 43rd North American Power Symposium, Boston, USA (2011)
12. Cvijić, S., Ilić, M.: Optimal Clustering for Efficient Computations of Contingency Effects in Large Regional Power Systems. In: IEEE PES General Meeting (2012)
13. Cvijić, S., Ilić, M.: Area-Level Reduction of Wheeling Loop Flows in Regional Power Networks. In: ISGT Europe 2012, Berlin (2012)
14. Patel, B.C.: Synthesis of Multiarea Grid Power Systems. PhD thesis, New Jersey Institute of Technology (1979)

Appendix

A System Parameters

The IEEE 14-bus, Figure 2, is used to illustrate the proposed algorithm for distributed DC power flow. Line parameters are listed in Table 1 while nodal injections are given in Table 2.

Table 1. Line parameters

From	To	R	X	B
1	2	0.01938	0.05917	0.0528
1	5	0.05403	0.22304	0.0492
2	3	0.04699	0.19797	0.0438
2	4	0.05811	0.17632	0.034
2	5	0.05695	0.17388	0.0346
3	4	0.06701	0.17103	0.0128
4	5	0.01335	0.04211	0
4	7	0	0.20912	0
4	9	0	0.55618	0
5	6	0	0.25202	0
6	11	0.09498	0.1989	0
6	12	0.12291	0.25581	0
6	13	0.06615	0.13027	0
7	8	0	0.17615	0
7	9	0	0.11001	0
9	10	0.03181	0.0845	0
9	14	0.12711	0.27038	0
10	11	0.08205	0.19207	0
12	13	0.22092	0.19988	0
13	14	0.17093	0.34802	0

Table 2. Generation and load

Bus Number	Generation [pu]	Load [pu]
1	2.324	0
2	0.400	0.217
3	0	0.942
4	0	0.478
5	0	0.076
6	0	0.112
7	0	0
8	0	0
9	0	0.295
10	0	0.090
11	0	0.035
12	0	0.061
13	0	0.135
14	0	0.149