(c) 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The definitive version of this paper is published in Proc. of IEEE SmartGridComm, 2013.

Confidentiality-preserving Obfuscation for Cloud-based Power System Contingency Analysis

Ognjen Vuković   György Dán
KTH Royal Institute of Technology
Stockholm, Sweden
Email: {vukovic,gyuri}@kth.se

Rakesh B. Bobba
University of Illinois at Urbana-Champaign
Urbana, IL, USA.
Email: rbobba@illinois.edu

***Abstract*—Power system operators are looking to adopt and migrate to cloud technologies and third-party cloud services for customer facing and enterprise IT applications. Security and reliability are major barriers for adopting cloud technologies and services for power system operational applications. In this work we focus on the use of cloud computing for Contingency Analysis and propose an approach to obfuscate information regarding power flows and the presence of a contingency violation while allowing the operator to analyze contingencies with the needed accuracy in the cloud. Our empirical evaluation shows, i) that the errors introduced into power flows due to the obfuscation approach are small, and ii) that the RMS errors introduced grow linearly with the magnitude of obfuscation.**

I. Introduction

Power grids around the world are undergoing a transformation to accommodate more renewable generation, allow consumer interaction with the infrastructure, and improve efficiencies through modernization. At the heart of this transformation are new sensor deployments, such as smart meters and phasor measurement. These new sensors are producing large volumes of data that a power system operator has to process and store, and increasing the number of devices that a power utility has to connect to and manage. To manage the data and connectivity to these devices utilities are looking to cloud based services. Smart meters in particular, given their numbers (in Millions even for medium size Utility) and geographic distribution, pose a challenge. Responding to this demand, many companies (*e.g.,* GE's GRID IQ, Honeywell's Akuacom, AutoGrid *etc.*) are offering cloud-based software-as-a-service models to manage smart meters and associated applications such as automated Demand Response (DR). Apart from customer facing applications such as Demand Response, utilities are also looking into leveraging cloud computing for other services such as managing security of their infrastructure as evidenced by the new CIGRE working group (D2.37) on cloud technologies for managed security [1]. The primary drives towards cloud computing are lower costs, improved efficiencies and elasticity of computing provided.

Power system applications related to operations, such as Contingency Analysis, forecasting, Optimal Power Flow, *etc.*, could also benefit from the advantages cloud technologies provide [2]). Security and reliability concerns are, however, a major barrier for adopting cloud technologies for power system operations [2], [3], especially with third-party providers. Recent work has addressed this issue from two sides. First, by improving the reliability and security provided by the cloud infrastructure for power grid applications, as is being done in the GridCloud [4], [5] project. Second, by transforming power system applications to preserve security properties such as confidentiality, integrity and availability in third party infrastructures. Borden *et al.,* [6] focus on transforming the optimal power flow problem before instantiating it in the cloud to preserve confidentiality.

In this paper we focus on contingency analysis (CA), which is a core application in power system operation. A contingency corresponds to the failure of one or more system components, such as a transmission line, a transformer, or a generator. The failure of any of these components would lead to a change in the power flows on the transmission lines, and could potentially result in an unstable system (e.g., power flows that exceed the thermal capacity of transmission lines). The aim of contingency analysis is to determine whether the power system would be unstable in case any of a potentially large set of contingencies would happen.

Contingency analysis is performed in modern energy management systems every time a new state estimate becomes available as a result of state estimation - as often as every few minutes. The number of contingencies that needs to be considered depends on the instantaneous load of the power system, the higher the load the more contingencies might have to be considered. The number of contingencies considered in practice is limited by the computational power available in the control center, and is often constrained to considering the loss of single components known as $N-1$ security. Cloud-based contingency analysis could allow an operator to scale the number of considered contingencies freely as a function of the instantaneous system state and enable $N-x$ security that is considered desirable, but it could expose the current system state and possible critical contingencies, thereby facilitating targeted attacks.

In this paper we propose an algorithm to obfuscate information regarding power flows and the presence of a contingency violation while allowing the operator to analyze contingencies with the needed accuracy in the cloud. We show that our approach doesn't introduce any error for CA using DC model. Further our empirical evaluation shows that the error introduced by the approach when using an AC model is quite small and that RMS error grows linearly with the magnitude of obfuscation applied.

The rest of the paper is organized as follows. Section II provides necessary background on contingency analysis. Section III presents our adversary model and usage scenario and

Section IV describes our obfuscation approach. Section V discusses some preliminary evaluation results and Section VI concludes the paper.

II. Background

We consider a power system that consists of N buses. We denote by P_n, $1 \leq n \leq N$ the power injection (load or generation) at bus n, and P_I is the vector of power injections. We denote the state of the power system by x. For simplicity, we consider active power flows only, in which case the system state is determined by the phase angles at the buses, and thus x is the vector of phase angles.

Given the system state x, the power flow between buses n and m can be computed as

$$P_{nm} = V_n V_m (G_{nm} \cos x_{nm} + B_{nm} \sin x_{nm}) = f_{nm}(x_{nm}), \tag{1}$$

where $x_{nm} = x_n - x_m$ is the phase angle difference between buses n and m, and G_{nm} and B_{nm} are the real and imaginary parts of the bus admittance matrix corresponding to buses n and m. The power injections can be computed using Kirchhoff's nodal law, and we denote the power injections as a function of the system state by $P_I = f_I(x)$. Finally, one can express the vector of power injections and power flows as a function of the system state as $P = f(x)$

A. AC Load-flow based Contingency Analysis

Let c be a contingency (e.g., the failure of two transmission lines), and let f^c be the function that describes the power flows under contingency c as a function of the system state, i.e., $P^c = f^c(x)$. Observe that a contingency might change the system topology and thus $f^c(.) \neq f(.)$. Similarly, the vector of power injections P_I^c under contingency c might be different from P_I, e.g., if the contingency involves the loss of one or more generators. To describe the relationship between the power injections before and after the contingency we introduce the fault matrix F^c such that $P_I^c = F_I^c P_I$. If contingency c does not affect the power injections then F_I^c is the identity matrix.

Given the vector of power injections P_I^c under contingency c, contingency analysis requires the solution of the load-flow problem, i.e., finding the state vector x^c that solves $P_I^c = f_I^c(x^c)$. The state vector is obtained through solving the power balance equations,

$$\Delta P_n \overset{d}{=} -P_n + \sum_m P_{nm} = 0. \tag{2}$$

Since the sum of the injections over all buses is zero, there are in total $N-1$ power balance equations and $N-1$ unknowns, as the phase angle of the reference bus is set to zero.

The equations (1) are non-linear, thus the solution to (2) is obtained using an iterative numerical method, typically the Newton-Raphson method [7]. Starting from an initial guess $x^c(0)$, the Newton-Raphson method obtains an updated estimate at iteration k by computing

$$\Delta x^c(k+1) = -J_k^{-1} \Delta P_I(k), \tag{3}$$

where $J_k = \frac{\partial P_I}{\partial x}|_{x=x^c(k)}$ is the Jacobian evaluated at the most recent guess $x^c(k)$, and then letting $x^c(k+1) = x^c(k) + \Delta x^c(k+1)$. Observe that the Jacobian is a non-singular square matrix of size $(N-1) \times (N-1)$. The algorithm terminates when the power mismatch ΔP_I is below a certain threshold. Let x^c be computed system state under contingency c.

Given the system state x^c under the contingency, the power flows can be calculated as $P^c = f^c(x^c)$. If any of the power flows exceeds the capacity limit (e.g., thermal capacity) of the transmission line then the system is said to be in an insecure state, and a corrective action must be taken by the operator to move the system to a state in which no contingency results in a capacity violation.

III. Adversary Model and Scenario

A. Adversary Model

We assume that the adversary has knowledge about the topology of the system but that he doesn't have access to the current state of the system. That is he does not know what the instantaneous power injections and power flows are. This adversarial model is inline with the recent body of work on false data injection attacks (*e.g.,* [8]–[10]) where the adversary is assumed to have full or partial knowledge of the H matrix for a DC model.

The goal of the adversary is to find the current system state (flows and injections) so he can determine if there are any contingencies with critical violations. Correspondingly, the goal of the obfuscation algorithm is to mask the real power flows from the adversary and to hide the existence of a violating contingency.

B. Usage Scenario

As shown in Figure 1, when a power system operator wants to undertake CA he will create an obfuscation vector and send the system with obfuscated flows to the cloud for contingency analysis. On obtaining the result of the CA for the various contingencies, the operator performs a deobfuscation step to obtain the power flows and injections that correspond to the non-obfuscated (actual) system. While obfuscation is performed only once, deobfuscation is performed for every contingency. Nevertheless, much of the computation of the deobfuscation can be done a-priori for a particular system topology.

IV. Obfuscated Contingency Analysis

In the following we first introduce the proposed obfuscation algorithm. We then show that for DC load flow calculation the proposed obfuscation does not introduce an error.

A. Obfuscation Algorithm

Consider the known power injections P_I^c under a contingency c. If an adversary has access to the power injections P_I^c and the computed power flows P^c under the contingency, it can infer which part of the system is most critical for stability and could perform a targeted attack. It is therefore important to obfuscate the information exposed to a potential attacker.

In the following we propose an algorithm that limits the attacker's ability to infer potential system instability. We do so by obfuscating the system state on which contingency analysis

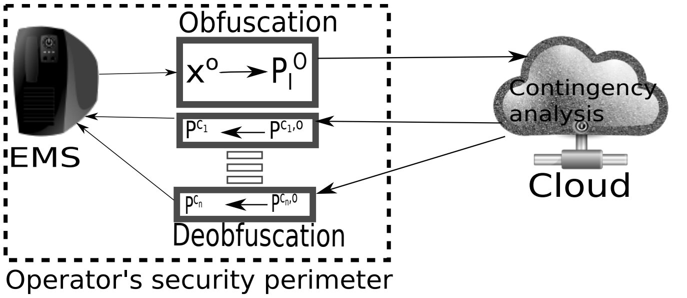
Obfuscation
$x^o \rightarrow P_I^O$
EMS
$P^{c_1} \leftarrow P^{c_1,O}$
$P^{c_n} \leftarrow P^{c_n,O}$
Deobfuscation
Operator's security perimeter
Contingency analysis
Cloud

Fig. 1. Considered scenario: Obfuscation is done once before contingency analysis is performed in the cloud, deobfuscation is done for all results.

is performed, and by compensating the contingent system for the modification after contingency analysis is performed. The important property of the proposed algorithm is that the computational cost of the obfuscation and of the deobfuscation is much less than that of the contingency analysis.

1) Obfuscation: Given P, the actual power flows in the system, obfuscation consists of adding a randomly chosen vector of power flows to the actual power flows. We refer to the latter as the *power flow obfuscation* vector,

$$P^o = Hx^o, \tag{4}$$

where $H = \frac{\partial P}{\partial x}$ is the Jacobian evaluated at the most recent system state (prior to the CA), and x^o is a non-negative vector of phase angles, the *state obfuscation* vector. We will discuss in Section IV-C how to create the state obfuscation vector. We use the state obfuscation vector to create the obfuscated system state, $x^O = x + x^o$. The obfuscated system state can be used to compute the obfuscated power injections as

$$P^O = f(x^O) \tag{5}$$

The obfuscated system state, and the corresponding obfuscated power injections $P_I^O = f_I(x^O)$ are the basis for the contingency analysis performed in the cloud.

For a particular contingency c, the obfuscated power injections $P_I^{c,O}$ are created, and are used as the input to the non-linear load-flow problem. The solution to the load-flow problem, i.e., the result of the analysis for contingency c is the state $x^{c,O}$ of the obfuscated contigent system.

2) Deobfuscation for a Contingency: Given the result $P^{c,O} = f_c(x^{c,O})$ of the contingency analysis performed on the obfuscated power flows for contingency c, deobfuscation consists of compensating for the power flows introduced through obfuscation.

To describe deobfuscation we define $H_c = \frac{\partial P^c}{\partial x}$, the Jacobian of the system under contingency c evaluated at the most recent system state (as in (4)). The deobfuscated power flows under contingency c are then obtained as

$$\tilde{P}^c = P^{c,O} - H_c {J_c}^{-1} P_I^{c,o}, \tag{6}$$

where $P_I^{c,o}$ is the vector of obfuscation power injections under contingency c. Note that if the contingency involves the loss of a generator then at least one or two entries in $P_I^{c,o}$ are changed and thus $P_I^{c,o} \neq P_I^o$.

Due to the non-linearity of the power balance equations, obfuscation will introduce an error in the result of the contingency analysis. We quantify this error by the difference of the power flows under a contingency with and without obfuscation

$$e_P = P^c - \tilde{P}^c. \tag{7}$$

To express the relative error we furthermore define the maximum componentwise relative error

$$\epsilon_P = \max_i \frac{e_{P,i}}{P_i^c}, \tag{8}$$

where P_i^c is the i^{th} component of the vector P^c.

B. Correctness under DC Load Flow-based CA

In the following we consider DC load flow computation and show that if contingency analysis is performed using DC load flow then the proposed obfuscation algorithm does not affect the result of the contingency analysis, i.e., the error e_P is zero.

The DC load flow model is based on the observation that in a system in normal operation the angular separation along any transmission line is small. This allows one to obtain a linear approximation for (1) of the form

$$P_{nm}^{DC} = V_n V_m (B_{nm} x_{nm}), \tag{9}$$

If one further considers that the per-unit voltages are approximately equal to one, then the power balance equations can be written as

$$\Delta P_n^{DC} \overset{d}{=} -P_n + \sum_m B_{nm} x_{nm} = 0. \tag{10}$$

Observe that due to the linearity of the power balance equations in the DC power flow model, the load flow problem for power injection vector P_I can be solved as $x = J^{-1} P_I$.

Proposition 1: Under DC load flow based contingency analysis the error introduced through obfuscation $e_P = \mathbf{0}$, where $\mathbf{0}$ is the vector of all zeros.

Proof: Consider the error e_P introduced by obfuscation in the result of the contingency analysis, as defined in (7),

$$\begin{aligned} e_P &= P^c - \tilde{P}^c \\ &= P^c - (P^{c,O} - H_c {J_c}^{-1} P_I^{c,o}) \\ &= H_c {J_c}^{-1} F_I^c J x - (H_c {J_c}^{-1} P_I^{c,O} - H_c {J_c}^{-1} F_I^c P_I^o) \\ &= H_c {J_c}^{-1} F_I^c J x - (H_c {J_c}^{-1} F_I^c (P_I + P_I^o) - H_c {J_c}^{-1} F_I^c P_I^o) \\ &= H_c {J_c}^{-1} F_I^c J x - (H_c {J_c}^{-1} F_I^c (Jx + Jx^o) - H_c {J_c}^{-1} F_I^c J x^o) \\ &= \mathbf{0}, \end{aligned}$$

where ${J_c}^{-1} P_I^O = x^O$ because of (4). ■

Note that the proof relies on the linearity of the power balance equations in the DC model, which implies that the DC load-flow problem can be solved in one iteration. Thus, the proof does not hold for AC load-flow based contingency analysis.

C. Choosing the Obfuscation Vector

In order to make obfuscation suitable for AC load-flow based contingency analysis, the choice of the obfuscation vector should be such that obfuscation does not introduce a significant error in the result of the contingency analysis, thus obfuscation should not be too big. At the same time, obfuscation should be big enough to hide the actual power flows from an attacker in the following sense. On the one hand, it should be ambiguous for an attacker whether a contingency exists in the actual system in case a critical contingency exists in the obfuscated system. On the other hand, if there is no critical contingency in the obfuscated system, the attacker can be aware of that there is no critical contingency in the actual system either, as this information cannot be used against the system.

The above two requirements imply that the power flow obfuscation P^o has to be bounded, and the obfuscation should have maximal entropy. We use the following result from [11] to construct the maximum entropy distribution.

Lemma 1: Fix real numbers $a < b$ and $\mu \in (a, b)$. The continuous probability density function on the interval $[a, b]$ with mean μ that maximizes entropy among all such densities (on $[a, b]$ with mean μ) is a truncated exponential density

$$q_\alpha(x) = \begin{cases} C_\alpha e^{\alpha x} & if x \in [a,b] \\ 0 & otherwise \end{cases} \tag{11}$$

where C_α is chosen so that $\int_a^b C_\alpha e^{\alpha x} dx = 1$, and α is the unique real number such that $\int_a^b x C_\alpha e^{\alpha x} dx = \mu$.

For $\alpha = 0$ the distribution is uniform on $[a, b]$, and its differential entropy is

$$h(X) = \int_a^b q_0(x) \log q_0(x) dx = \log(b - a). \tag{12}$$

Proof: We refer to [11] for the proof. ■

As our objective is to obfuscate the power flows, we define the obfuscation vector in terms of the obfuscation power flows P^o, and use the uniform distribution for obfuscation. We thus define the diagonal matrix U with diagonal elements $U_{i,i} \sim U(0, 0.1)$, and create the vector

$$\hat{P}^o = UP. \tag{13}$$

This vector cannot be used directly for the obfuscation because it does not necessarily correspond to any system state. We therefore perform a linearized state estimation on this vector to obtain the state obfuscation vector

$$x^o = (H^T H)^{-1} H \hat{P}^o, \tag{14}$$

where H^T is the transpose of H. Note that the components of x^o do not follow a uniform distribution, but the components of the power flow obfuscation vector $P^o = Hx^o$ are likely close to uniform (relative to the actual power flows). Numerical results presented in Section V show that this is indeed the case.

V. Performance Evaluation

In the following we illustrate the efficiency of the proposed algorithm via simulations.

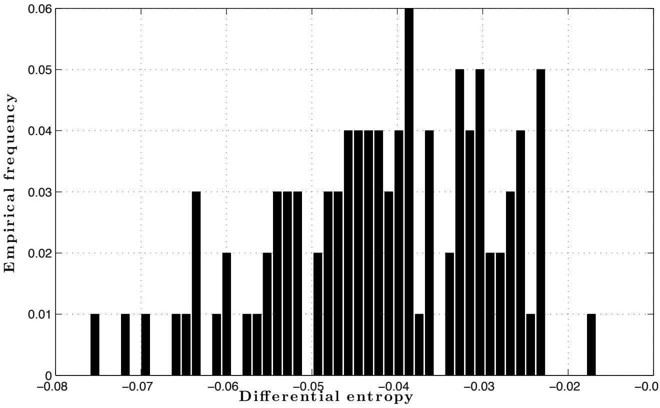

Fig. 2. Histogram of the differential entropy of the relative obfuscation power flows for $u_{max} = 0.1$, computed over 100 runs.

A. Simulation methodology

We used the IEEE 118 bus test system and and used Matpower for the AC load flow based CA. The power flows and injections are represented using p.u, where 1 p.u. equals to 100MW. For the obfuscation, we considered all active power injections and all active power flows, both 'to' and 'from' buses (hence negative values in the Figures).

B. Obfuscation performance

We first consider the performance of the algorithm in terms of the obfuscation it provides. Note that the level of obfuscation does not depend on the particular contingency considered, it depends on the system topology and the actual system state. These results are thus general for the IEEE 118 bus system.

Figure 2 shows a histogram of the differential entropy of the relative obfuscation power flows, i.e., that of P^o/P in a component-wise sense, computed over 100 randomly chosen obfuscation power flows for $u_{max} = 0.1$. We approximated the differential entropy by creating a histogram with 200 bins and using the histogram bins width for numerical integration. The differential entropy of $U(0, 0.1) \approx -3.2$, thus aligning the power flows with the range space of the Jacobian in (14) does alter the distribution of the power flows, but it does not decrease its entropy. In fact, the obfuscation of some power flows by far exceeds $u_{max} = 0.1$, which is the reason for the significantly higher entropy than with the uniform distribution.

Figure 3 shows the QQ plot of the distribution of $P^o = Hx^o$ defined in (14) normalized by P, compared to a uniform distribution on $[0, 0.1]$, computed over 100 randomly chosen obfuscation vectors. Recall that the components of $\hat{P}^o$ follow a uniform distribution, but due to (14) the components of P^o do not necessarily do so. Figure 3 shows that the distribution of P^o indeed differs from uniform, especially at the tails, which is also the reason for the increased differential entropy, as discussed above. At the same time, the body of the distribution is close to uniform. The figure also shows that there is not much difference between the individual obfuscation vectors, as the percentiles are rather close to the average.

These two figures indicate the choice of the obfuscation vector P^o provides a good level of randomness thus making it

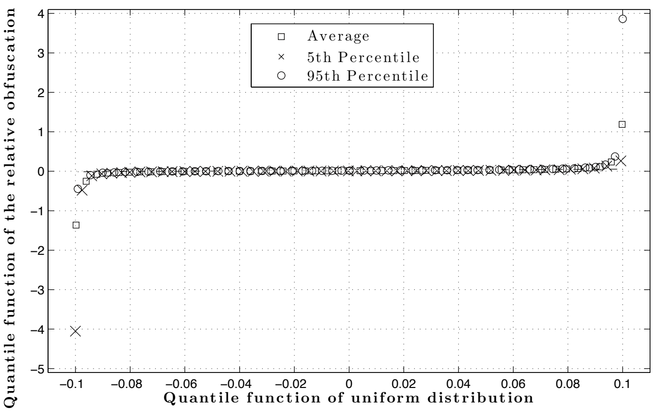

Fig. 3. QQ plot of the distribution of the relative obfuscation power flows and injections P^o/P for $u_{max} = 0.1$ vs. a uniform distribution $U(0, 0.1)$, computed over 100 runs.

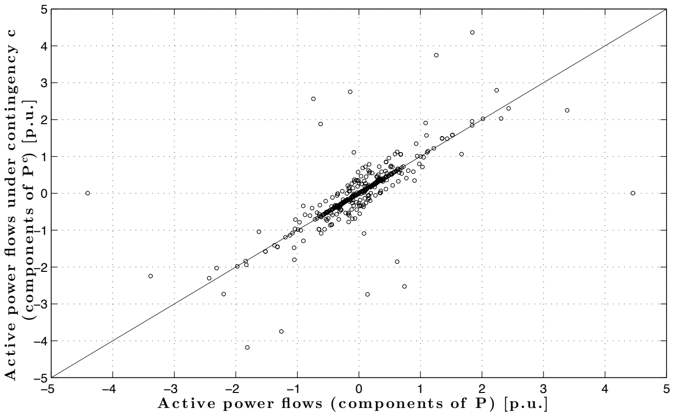

Fig. 4. Active power flows after the contingency vs. before the contingency.

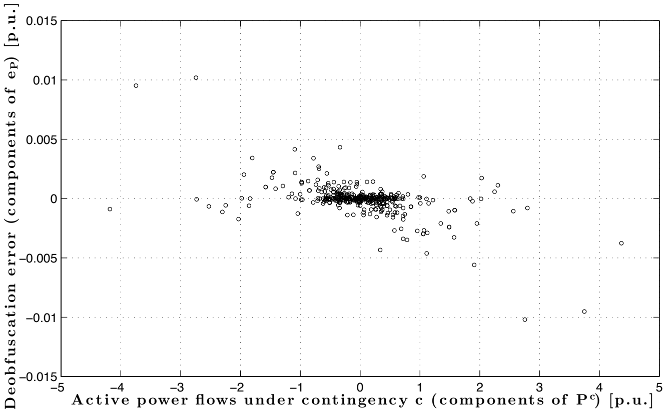

Fig. 5. Error introduced by obfuscation for $u_{max} = 0.1$ vs. the power flows under contingency obtained with regular CA.

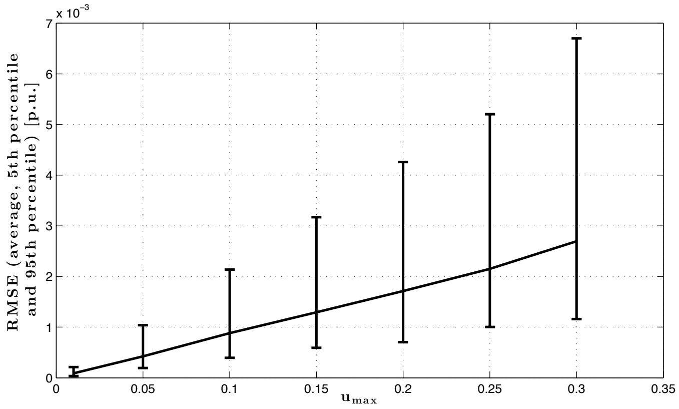

Fig. 6. Impact of domain of the uniform distribution $U(0, u_{max})$ on the average root mean square error of obfuscated CA, mean, 5 and 95 percentiles.

hard for an adversary to guess the real power flows. We now turn to the evaluation of the error introduced by obfuscation for AC load flow-based CA.

1) Obfuscation vs. CA accuracy: In the following we consider a contingency that affects branch 9, which is a transmission line that connects buses 9 and 10. The effect of the contingency on active power flows is shown in Figure 4. The figure shows that the pre-contingency power flow on branch 9 is above 4 p.u., and is the largest power flow in the system, thus the scenario corresponds to a severe contingency.

Figure 5 shows a scatter plot of the error vs. the power flows under the considered contingency after deobfuscation: the power flows P^c obtained without the proposed scheme are shown on the horizontal axis, and the errors e_P remaining in the corresponding power flows $\hat{P}^c$ after deobfuscation are shown on the vertical axis. Thus, every dot shown corresponds to an error in a power flow or a power injection. The results shown were obtained for $u_{max} = 0.1$. The figure shows that the errors introduced by obfuscation are small, all dots are located close to zero, which corresponds to no error, i.e., $e_P = \mathbf{0}$. The figure thus shows that the errors are very small compared to the actual power flows.

Figure 6 shows the average root mean square error (RMSE) introduced in the result of the CA by obfuscation as a function of the upper bound u_{max} of the uniform distribution used for obfuscation in (13). The average RMSE is defined as $\frac{||e_P||_2}{|e_P|}$, where $|.|$ is the number of components in the vector. For every u_{max} value the figure shows the mean over 100 simulations together with the 5 and 95 percentiles. The figure shows that the average RMSE increases approximately linearly over a wide range of u_{max} values, and so do the percentile values. The average RMSE is very small compared to the actual power flows in the system, which confirms that obfuscated CA would be viable.

Figures 7 and 8 show the difference between the obfuscated power flows $P^{c,O}$ and the power flows obtained without obfuscation P^c for two different obfuscation vectors x^o, but only for those power flows that increase due to the contingency. The vertical axis is thus effectively the introduced obfuscation. Both figures show that the amount of obfuscation grows with the power flow, but the actual values differ because they depend on the obfuscation vector x^o. The two obfuscation vectors used for Figure 7 and for Figure 8 were chosen from the considered 100 obfuscation vectors for $u_{max} = 0.1$ so as to represent two different scenarios in terms of the signs of the introduced obfuscations per flow. In the first scenario (Figure 7), all power flows that increased due to the contingency without obfuscation

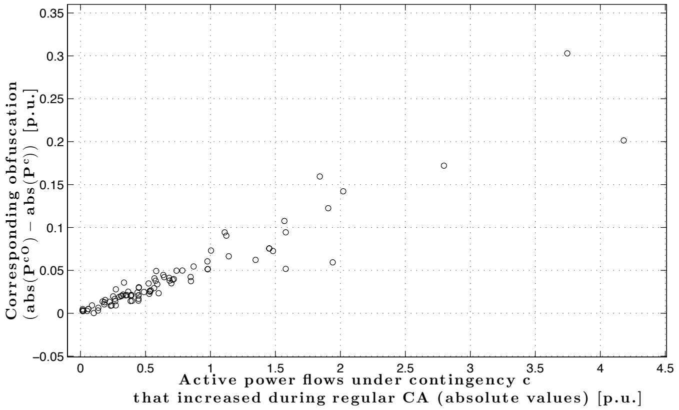

Fig. 7. Obfuscated power flows under the contingency vs. power flows that increased due to the contingency with the regular CA. All obfuscated power flows exceed the actual power flows.

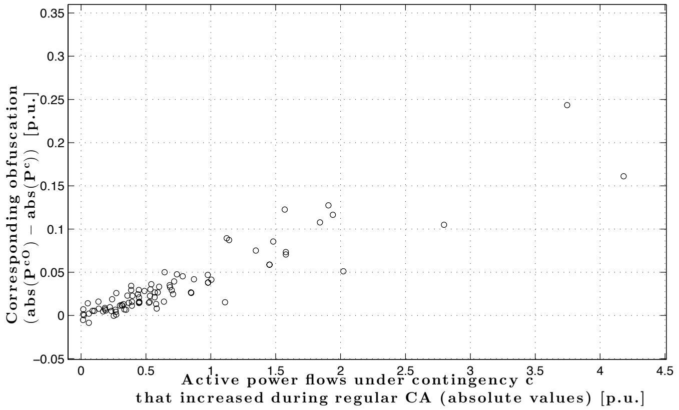

Fig. 8. Obfuscated power flows under the contingency vs. power flows that increased due to the contingency with the regular CA. Most of the obfuscated power flows exceed the actual power flows.

have a positive amount of obfuscation, while in the second scenario (Figure 8), there are a few relatively small power flows for which the obfuscation is negative. Power flows that have a negative obfuscation are determined by the obfuscation vector x^o, which is unknown to the attacker. Consequently, by just observing $P^{c,O}$, an attacker cannot be certain how much obfuscation is introduced and for which flows the obfuscation is negative. Thus, the fact that there is a thermal capacity violation in the obfuscated system does not imply that it is also the case after de-obfuscation, and thus an attacker that observes a violating contingency based on $P^{c,O}$ cannot be certain that there is a violating contingency in the actual system, according to P^c.

VI. Conclusion

We proposed an approach to obfuscate information regarding power flows to enable CA in the cloud while allowing the operator to obtain accurate post contingency flows. Our approach doesn't introduce any error for CA using a DC model and our numerical results show that the error introduced when using AC models is tolerable. It is subject of our future work to extend the obfuscation algorithm so that it always introduces positive obfuscation to the power flows that increase due to contingency. Furthermore, our future work will include analytically bounding the error introduced by the proposed obfuscation and an analytical characterization of the randomness of the obfuscation vector.

Acknowledgements

This material is based upon work supported in part by the Department of Energy under Award Number DE-OE0000097[1].

References

[1] [Online]. Available: http://d2.cigre.org/WG-Area/D2.37-Guidelines-for-outsourcing-managed-security-services-using-Cloud-Technologies

[2] G. Dan, R. B. Bobba, G. Gross, and R. H. Campbell, "Cloud Computing for the Power Grid: From Service Composition to Assured Clouds," in *In Proceedings of 5th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '13)*, June 2013.

[3] K. P. Birman, L. Ganesh, and R. van Renesse, "Running smart grid control software on cloud computing architectures," April 2010.

[4] K. Maheshwari, M. Lim, L. Wang, K. Birman, and R. van Renesse, "Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud," *IEEE PES Innovative Smart Grid Technologies*, 2013.

[5] K. Maheshwari, K. Birman, J. M. Wozniak, and D. Van Zandt, "Evaluating cloud computing techniques for smart power grid design using parallel scripting," in *IEEE/ACM International Symposium On Cluster, Cloud And Grid Computing (CCGrid)*, 2013.

[6] A. R. Borden, D. K. Molzahn, P. Ramanathan, and B. C. Lesieutre, "Confidentiality-preserving optimal power flow for cloud computing," in *Allerton Control Conference*, 2012.

[7] B. Stott, "Review of load-flow calculation methods," *Proc. of the IEEE*, vol. 62, no. 7, pp. 916–929, 1974.

[8] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions in Information and Systems Security (TISSEC), 2011*, vol. 14, no. 1, pp. 13:1–13:33, June 2011.

[9] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on dc state estimation," in *1st Workshop on Secure Control Systems (SCS '10)*, 2010.

[10] G. Dán and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in *First IEEE International Conference on Smart Grid Communications (SmartGridComm), 2010*, Oct. 2010, pp. 214 –219.

[11] K. Conrad, "Probability distributions and maximum entropy," 2012. [Online]. Available: http://www.math.uconn.edu/~kconrad/blurbs/analysis/entropypost.pdf

[1] This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.