

## RESEARCH STATEMENT

**Denis Ulybyshev,**

Research Assistant, Ph.D. Candidate;

Department of Computer Science, CERIAS, Purdue University

Cybersecurity Software Engineer, Coze Health LLC

Email: [dulybysh@purdue.edu](mailto:dulybysh@purdue.edu)

Homepage: <https://cs.purdue.edu/~dulybysh>

### Secure Data Exchange and Leakage Detection

**Keywords:** Privacy, Access Control, Cloud Data Management, Data Leakage Detection, Encrypted Search, Blockchain-based Technologies

**Abstract:** My philosophy in research is to solve practical, real-world and challenging problems in cybersecurity, databases and blockchain-based technologies. I develop and implement practical techniques and prototypes.

### Problem Statement

My research aims to provide the following:

1. Data protection at rest and in transit, providing data leakage prevention and detection with threat assessment as well as enforcing role-based and attribute-based access control.
2. Encrypted search over encrypted data, with capabilities of performing data analysis over encrypted data.
3. Collaborative secure software development system which collects provenance data and provides its integrity.
4. Targeted Information Propagation based on machine learning algorithms.

### Data protection at rest and in transit

In service-oriented architecture (SOA), services can communicate and share data among themselves. Services and associated data can be hosted by cloud platforms, which are vulnerable to large attack surface that could violate data privacy. My colleagues and I have designed a solution that provides data protection in transit and at rest. This solution also provides data leakage prevention and detection for multiple leakage scenarios that can be performed by an external attacker or a malicious insider. The prototype called “WAXEDPRUNE” (Web-based Access to Encrypted Data Processing in Untrusted Environments), implemented in collaboration with Northrop Grumman, MIT and W3C, was demonstrated at Northrop Grumman Tech Expo in 2016. The approach ensures that each service can access only those data subsets for which the service is authorized. Encrypted search and extensive data analysis over encrypted data records are supported as well. “WAXEDPRUNE” project received two awards. In April 2017, it was selected to be funded as #1 out of 21 research projects by Corporate Partners of Computer Science Department and CERIAS at Purdue University, including Northrop Grumman, Intel, Qualcomm, Raytheon, Eli Lilly. In March 2015, the research poster “PD3: Policy-based Distributed Data Dissemination”, which is based on the predecessor of “WAXEDPRUNE”, was selected by Corporate Partners as #1 out of 43 at 16th CERIAS Security Symposium.

Data protection method is based on using Active Bundles (AB). AB is a self-protecting structure (see Fig.1) that contains data in encrypted form, access control policies and a policy enforcement engine (Virtual Machine). Data are stored in AB in the form of key-value pairs with encrypted values. Each data subset is encrypted with its own symmetric key using a novel “on-the-fly” key generation scheme, based on the execution flow. As a

use case, Electronic Health Record (EHR) of a patient can be stored as an AB. An example of a key-value pair stored in an AB, which represents EHR, is:  $\{ "ab.patientID" : "Enc(0123456)" \}$ .

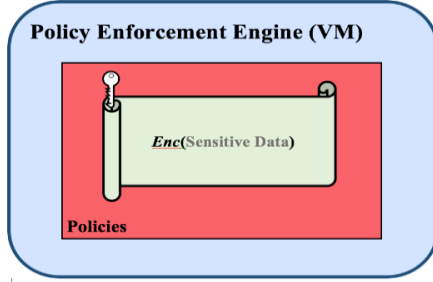


Fig.1. Active Bundle

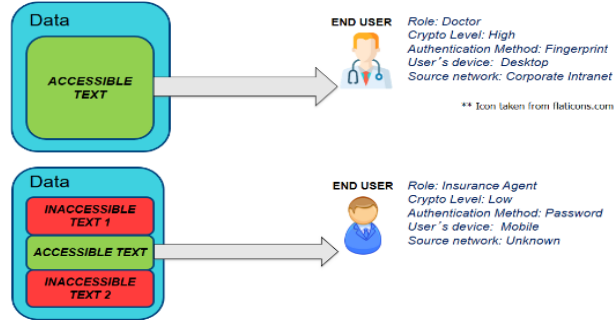


Fig.2. Accessible datasets for different roles and attributes

Patient ID = 0123456 is stored in the AB in an encrypted form. When a service requests data from an AB, the identity of the requesting service is verified as the first interaction step. This authentication is based on signed digital certificates. The services present their X.509 certificates signed by a trusted Certificate Authority (CA) to the AB to verify their authenticity. Then, if the authentication passes, access control policies, service attributes and context by an AB kernel. Evaluated attributes include:

- 1) Client's authentication method (password-based vs. hardware-based vs. a fingerprint). A password-based authentication method is considered to be least secure.
- 2) Level of cryptographic capabilities of a client's browser.
- 3) Type of the client's device (Mobile vs. Desktop).
- 4) Type of the network (e.g. unknown vs. corporate intranet).
- 5) Trust level of a client, which is constantly recomputed, based on the following metrics: (a) number of sent/rejected data requests, (b) CPU/Memory usage, (c) number of communication failures.

Based on these evaluations, symmetric decryption keys are generated to decrypt those data items for which the authenticated service is authorized (see Fig. 2). The symmetric key generation is based on the unique information generated in the execution control flow path of an AB. This information depends on the AB modules and their resources: the authentication code, the subject's role (which is extracted from the X.509 certificate of the subject, and which can be a doctor, an insurance agent, a researcher, etc), the authorization code, the applicable access control policies and the policy evaluation code. AB is tamper-resistant and protects from an attacker who tries to:

- (a) modify AB code to bypass authentication phase or evaluation of access control policies and client's attributes;
- (b) modify access control policies to gain access to data that the attacker is not authorized for;
- (c) impersonate service identity by using the wrong X.509 certificate to gain unauthorized access to data.

Modification of any of the items above will result in change of the digest and will lead to an incorrect decryption key derivation. The genuine digest is calculated when AB is created. This tamper-resistance mechanism provides data integrity and storing data in an encrypted form provides data confidentiality in untrusted environments.

*Adversary model:*

1. Cloud provider or server may have a curious or malicious administrator that tries to access confidential data or modify them.
2. Client can be malicious and try to:
  - a) leak data, for which client is authorized, to unauthorized parties;
  - b) modify the AB code or access control policies to gain unauthorized access to confidential data.

In my design, I address two leakage scenarios: leakage of the entire AB and leakage of decrypted plaintext data from AB, made by authorized party (insider) to unauthorized one. Data in leaked AB are stored in encrypted form and to extract the data attacker will need to break the AES encryption algorithm. Access to unauthorized data will be denied by the AB kernel. An attempt to decrypt data made by an unauthorized service will be recorded by a trusted Central Monitor (CM). Every AB transaction in the data exchange network is monitored by the CM, which is notified each time a client tries to decrypt a data subset from an AB. A notification message contains information on what service attempts to decrypt what type of data, when, who is the origin or sender of the AB. The problem to detect leaking service is similar to “traitor tracing” problem, discussed by Dan Boneh<sup>1</sup>. The CM queries its local database of access control policies, in order to check whether the service that tries to decrypt data is authorized for that class of data. If not, then multiple actions can be taken:

- a) trust level for Sender and Receiver is decreased;
- b) requested data subset  $d_i$  is marked as compromised and all other services are notified about this;
- c) AB is re-created with stricter access control policies to make it stronger against similar leakages. The process includes the following:
  - c1) separating the compromised role of a Sender into *Role* and *Trustworthy\_Role*, e.g. “*Doctor*” and “*Trustworthy\_Doctor*”;
  - c2) sending new certificates with *Trustworthy\_Role* to all trustworthy subjects with the previous *Role*;
  - c3) creating a new AB with modified policies to prohibit data access for *Role*;
  - c4) disabling the “Save As” functionality to prohibit storing sensitive data locally;
  - c5) raising the sensitivity levels for leaked data types to prevent leakage repetition.

Without obtaining permission from the trusted CM, the data decryption process will not start. Furthermore, data requesting service might be asked to get an activation code from the authentication server, which is under CM’s control, and which will notify the CM that certain data item  $d_i$  has arrived, from where and to whom. In addition to the data access control policies enforced by CM, leakage detection relies on a web crawler to verify digital watermarks that are embedded into AB. If an AB is uploaded to a publicly available network directory, then the crawler verifies its digital watermark to check whether the AB is supposed to be at that network node. We assume that it is possible to determine the identity of the node that hosts a public directory (e.g., in the Hospital Intranet). Network nodes, which participate in data exchange, use X.509 certificates, which identify their roles (e.g., a doctor or insurance agent or researcher). In addition, identity can be based on a node’s IP address or other attributes.

Second, a more challenging data leakage scenario that I address is when a service authorized for data  $d_i$  can get these data from an AB, store them locally in the plaintext form and then send them without the AB behind the scene to an unauthorized service as plaintext<sup>2</sup>. For instance, an authorized client (a malicious insider) can take a picture of a displayed  $d_i$  on a mobile phone’s camera when  $d_i$  is displayed on a computer screen. Protection provided by the AB is gone in this case, and we cannot prevent plaintext  $d_i$  leakage. We aim to help investigating the leakage and do forensics based on provenance records stored on CM each time a data request is served by the AB. The provenance records contain information on who is trying to access what class of data, when, and who is the origin/sender of this AB. To mitigate a leakage problem, we embed digital watermarks into data and visual watermarks on a web page where data retrieved by a client from an AB are displayed in the client’s browser. Digital watermarks are embedded into RGB images by means of the conversion function  $F(r, g, b)$  which is applied to every pixel of an image. It changes the RGB image in such a way that it is indistinguishable by the human eye from the original image. However, a web crawler with a built-in classifier is able to determine whether the RGB image has an embedded watermark or not. This watermarking method works only if the RGB image is stored in a publicly available folder. Once a watermark is detected, CM is notified, and it checks whether the given RGB image is supposed to be at a given network node. We assume that it is possible to determine the identity of the node that hosts a public directory or its role. A node identity can be based on node’s IP address or other attributes. Network nodes, which participate in data exchange, use X.509 certificates that identify their roles.

<sup>1</sup> D. Boneh, M. Franklin, “An efficient public key traitor tracing scheme”. In *Crypto* (Vol. 99, pp. 338-353), 1999.

<sup>2</sup> D. Ulybyshev, B. Bhargava, A. Alsalem, “Secure Data Exchange and Data Leakage Detection in Untrusted Cloud”, Springer Journal on 1-st Intl Conf. on Applications of Computing and Communication Technologies (ICACCT), 2018, pp. 99-113

There are two types of visual watermarks for displaying data: clearly visible ones and very small ones that are visible only if zoomed. The former will remain in the image if a malicious insider takes a picture of a screen. For some types of data, it is easy to reproduce the screen's content and write it down, e.g., on a piece of paper. It removes the visual watermark. However, for some types of medical information, (e.g. X-ray images), it is hard to reproduce them on a piece of paper but easy to take a picture of a screen, which contains visual watermarks.

I propose the following additional data leakage detection/prevention/alleviation methods:

1) *Partial Data Disclosure:*

- a) Authorized client after the first data request is only given a portion of accessible data;
- b) Monitoring the client's trust level, which is constantly re-computed by the Central Monitor using the following metrics: (a) number of sent/received/rejected data requests, (c) number of communication errors, (d) CPU/Memory usage;
- c) Disclose the next accessible chunk of data, provided trust level is satisfactory.

2) *"Fake" leakage*

In case of a detected data leakage, several other "fake" versions of data might be intentionally leaked to lower the value of the leaked data. For example, if a patient's address got leaked then false addresses for the same patient might be intentionally leaked.

3) *Elevation of a data classification level*

The idea is to raise the classification level for the class including the leaked data to prevent leakage repetition.

Data leakage damage is evaluated using the following information:

- 1) How malicious is the recipient of unauthorized data;
- 2) Sensitivity of data that got leaked;
- 3) Leakage timing
- 4) Inference threat, which indicates whether other data can be inferred from the data that got leaked

$$\text{Damage} = K_{ds} (\text{Data Sensitivity}) * K_{sm} (\text{Service Maliciousness}) * F(t) \quad (1)$$

$K_{ds}$  is data sensitivity coefficient,  $K_{sm}$  is the service maliciousness coefficient,  $F(t)$  is the data sensitivity function.

*Summary of important AB features:*

1. Provides data confidentiality and integrity
2. Provides role-based and attribute-based access control
3. Is implemented as a Java Executable Archive (JAR) and supports industrial standards, such as X.509.
4. Access control policies are specified using Javascript Object Notation (JSON).
5. Open-source WSO2 Balana is used for policy evaluation. AB is agnostic to policy enforcement engine.
6. Supports on-the-fly data updates for the authorized parties and allows redacting data.
7. Symmetric keys, used to encrypt/decrypt sensitive data, are not stored neither on a cloud provider nor inside an Active Bundle nor on any Trusted Third Party (TTP)
8. Can work in hybrid network architectures, supporting both centralized and decentralized data communications amongst web services.
9. Can be hosted by network nodes in peer-to-peer networks or by cloud providers that serve data requests.
10. Supports data leakage detection for multiple leakage scenarios.
11. Supports fast data analytics over encrypted data.

*Assumptions:*

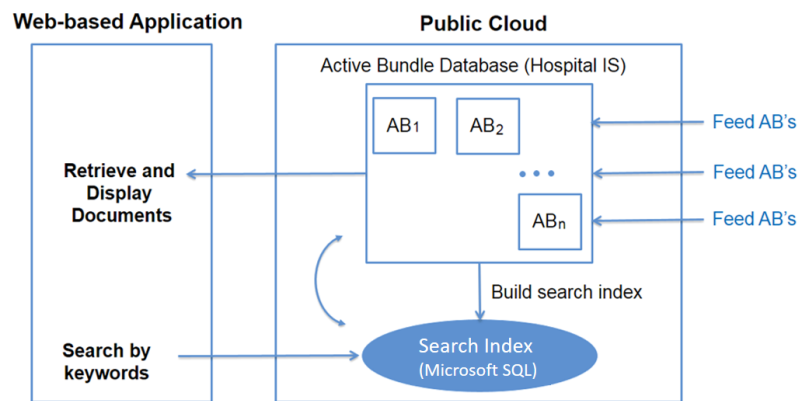
1. The entity that hosts/executes AB has trusted hardware, a trusted operating system and a trusted Java Virtual Machine.

*Note:* to relax this assumption, Intel SGX or ARM TrustZone architectures can be used

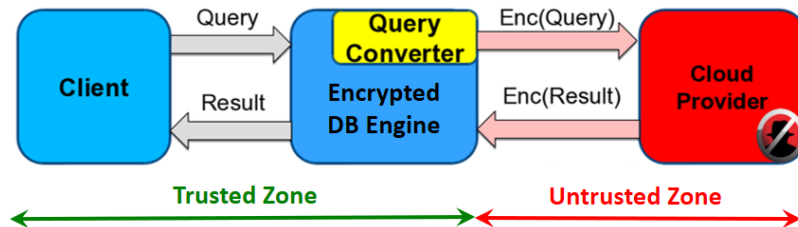
2. HTTPS protocol is used for data communications among all the web services.

## Encrypted Search over Encrypted Data

Public cloud provider can host a database of ABs. The database contains extra attribute(s) used for indexing, e.g. a short abstract of AB and its keywords (see Fig. 3). Extra attributes, as well as data inside AB, are stored in encrypted form. Unique ID maps each record, containing extra attributes, to its corresponding AB, e.g. to the Electronic Health Record of a given patient for healthcare use case. As it can be seen from Fig. 4, the user's search query is converted to a SQL query, then the SQL query is converted into its encrypted form. User-defined functions are used to do this conversion. The framework is agnostic to the encrypted database engine. I use Microsoft SQL Server, which provides data encryption and supports set of SQL queries to operate on encrypted data.



*Fig.3. Encrypted search over encrypted data (idea proposed by Dr. Leon Li, NGC)*



*Fig.4. Framework for encrypted search over encrypted records*

The essential design decision is to rely on two-layer approach in retrieving encrypted information from ABs<sup>3</sup>. In the first phase of data request, search query is executed over index database, which contains encrypted index attributes, employing Partially Homomorphic Encryption. Relevant ABs are retrieved using unique ID, which maps every index record to the corresponding AB. In the second phase, data request is served by only relevant ABs. Forwarding data request only to relevant ABs significantly improves the performance since multiple steps need to be executed by AB kernel to serve data request. These steps include client authentication and evaluation of access control policies, client's attributes and context. In addition to improving performance, this approach provides confidentiality of the database records. Furthermore, the approach inherits all the advantages of ABs, including role-based and attribute-based access control and detecting data leakages for multiple scenarios.

As a use case, AB can store the vehicle record, which contains owner's info, including name, address, phone number, driver license number; vehicle's info, including VIN number, health check results for engine, brakes, transmission, etc; and road information on traffic jams, accidents, obstacles, road constructions, etc. Index

<sup>3</sup> D. Ulybyshev, A. Alsalem, B. Bhargava, S. Savvides, G. Mani, L. Ben-Othmane, "Secure Data Communication in Autonomous V2X Systems", IEEE ICIOT, San-Francisco, 2018

database contains encrypted attributes, including speed, license plate number, timestamp of measuring speed and ID. Encrypted vehicle records, as well as encrypted index database, are stored in cloud. Intelligent Transportation System or Law Enforcement might need to get personal data of drivers who exceeded speed limit of 65 mph on a highway and drove above 95 mph. In the first phase, the required SQL query is:

```
SELECT ID FROM IndexDB WHERE SPEED > 95;
```

Converted query:

```
SELECT c1 FROM Alias1 WHERE ESRCH (Enc(Speed), Enc(95));
```

In the second phase, http(s) GET request for drivers' name, address and license number is sent to relevant ABs, i.e. to ABs with IDs returned by SQL query in the first phase.

AB may include an extra data field in encrypted form, e.g. "AB Summary", which is used to build fast data analytics. "Summary" field can be encrypted with MMCP key, which is derived based on captured attributes of a vehicle, including make, model, color and license plate. Convolutional neural networks technique is utilized. The service who has MMCP-key, e.g. analytics administrator, can decrypt "AB Summary" value from AB without going through policy evaluation<sup>4</sup>. It allows to perform fast on-the-fly data analytics.

## Blockchain-based secure software development system

To ensure *integrity, trust and immutability* of software and data (cyber data, user data, and attack event data), me and my colleagues designed blockchain-based technology<sup>5</sup>, "Blockhub", for collaborative secure software development. Every access, transfer and update of data and software is recorded in the blockchain public ledger, can be verified any time in the future and cannot be erased or repudiated by invokers. Blockchain-based technology allows tracking and controlling what data or software components are shared between entities across multiple security domains. "Blockhub" integrates "WAXEDPRUNE" solution. Same as in WAXEDPRUNE project, Active Bundle (AB) self-protected structure is used in Blockhub for data storage and transfer. In Blockhub prototype, AB structure is upgraded to a Software Bundle (SB), which, in addition to encrypted data, stores software source codes and executables in encrypted form. SB provides identity management, role-based and attribute-based access control as well as leakage detection capability. It guarantees that each party can access only those data subsets and software modules for which the subject is authorized.

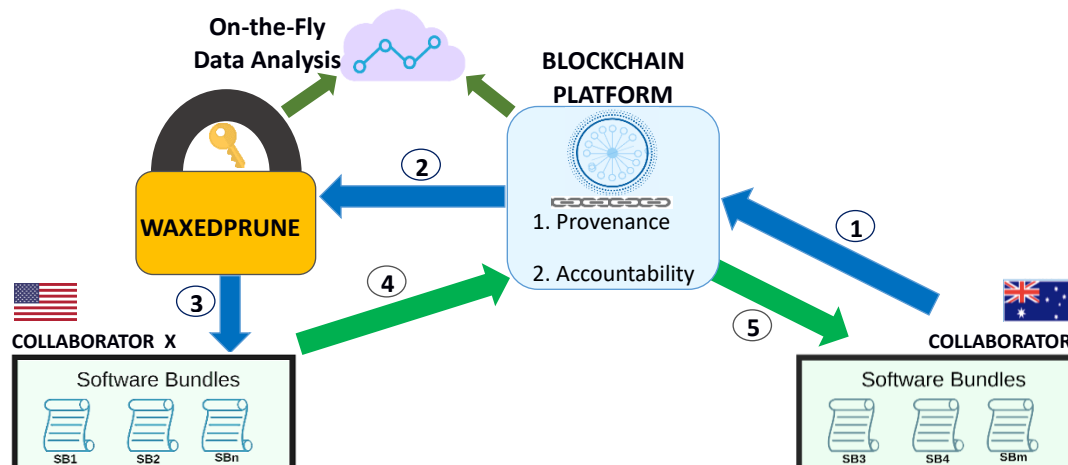


Fig. 5. Blockhub: Secure Software Development Framework

<sup>4</sup> D. Ulybyshev, S. Palacios, G. Mani, A. Alsalem, B. Bhargava, P. Goyal, "On-the-fly Analytics over Encrypted Records in Untrusted V2X Environments", ICACEEE 4-th Intl. Conf., Zurich, 2018

<sup>5</sup> D. Ulybyshev, M. Villarreal, B. Bhargava, G. Mani, S. Seaberg, P. Conoval, R. Pike, J. Kobes "Blockhub: Blockchain-based Software Development System for Untrusted Environments", IEEE CLOUD, San-Francisco, 2018



Two untrusted collaborators, e.g. X and Y (see Fig. 5), can securely share data and software in a controlled manner. Every software or data request is registered in blockchain network (see step 1 on Fig. 5), before being forwarded to the corresponding SB (step 2). On step 3, identity of the client is checked and SB kernel evaluates access control policies and client attributes, including trust level. On step 4, transaction result is recorded in the blockchain network and on step 5 the accessible data is sent to the client, using https protocol. Blockchain technology provides integrity of provenance data. It is used for effective data forensics/provenance in cross-domain data communication networks and in global software collaboration environments with multiple untrusted writers. SB, in turn, provides integrity of user data and software modules, including source codes and binary executables.

In the future, I plan to integrate Blockhub with GitHub or different software repository, so that every software access or update is recorded in the blockchain network. Furthermore, Blockhub will enable flexible role-based and attribute-based access control, as well as leakage prevention and detection.

## Targeted Information Propagation

The data from heterogeneous sources are extracted, transformed, consolidated, cleaned and then are fed into the Knowledge Discovery Engine, which applies supervised and unsupervised learning algorithms to detect data patterns relevant to user's needs. Algorithms, e.g. used by autonomous vehicles, include detection of objects, such as vehicles, road signs, parking spaces, pedestrians, and human face recognition. The results of the detection algorithms are pushed to the relevant subjects or relevant roles. Subject profiling algorithm identifies online which information the subject may need now to complete the task, including collaborative tasks. Initial input to the profiling algorithm<sup>6</sup> is an audit log data record from which the system administrator can select  $n$  most appropriate attributes. Having a certain number of such data records, a certain number of clusters can be formed and periodically updated. Also, the system is able to check whether the new subjects behave according to their roles and to detect anomalies. For role-based and attribute-based access control, "WAXEDPRUNE" project will be employed in the future.

## Funding

My main source of research funding comes from a corporate sector. Northrop Grumman funded the projects I have been working in for 5 consecutive years, renewing the funding every year, starting from 2014. Overall, I contributed to writing significant parts of five research proposals that succeeded to be funded. Our research group with my Academic Adviser, Professor Bharat Bhargava, collaborates with MIT, W3C, University of Western Michigan.

My research was also partially funded by Qatar National Research Fund (a member of Qatar Foundation).

In 2017, my research proposal won a grant of around \$45,000 (covering one year of tuition and stipend) from the Corporate Partners of Computer Science Department at Purdue University, including companies such as Northrop Grumman, Intel, Qualcomm, Raytheon, Eli Lilly.

From 2013 to 2014 I had been working for four semesters as a research assistant in interdisciplinary NSF project "Robust Distributed Wind Power Engineering" with Prof. Suresh Jagannathan, Prof. Jan Vitek and Prof. Ananth Grama as Principal Investigators. It involved collaboration between Computer Science and Mechanical Engineering departments and resulted in interdisciplinary publication<sup>7</sup>.

In addition to writing successful research proposals for a corporate sector, I have an experience in writing NSF, NIH and DARPA proposals on cybersecurity, blockchain-based technologies and data analysis projects.

---

<sup>6</sup> E. Terzi, Y. Zhong, B. Bhargava, Pankaj, and S. Madria, "An Algorithm for Building User-Role Profiles in a Trust Environment", Proc. Of 4-th Intl. Conf. Data Warehousing and Knowledge Discovery (DaWaK), vol. 2454, 2002.

<sup>7</sup> N. Myrent, D.Adams, G.Rodriguez-Rivera, D. Ulybyshev, J.Vitek, E.Blanton, T. Kalibera, "A Robust Algorithm to Detecting Wind Turbine Blade Health Using Vibro-Acoustic Modulation and Sideband Spectral Analysis", 33rd ASME Wind Energy Symp., 2014