PUR-CM-18-228

ACQUISITION RESEARCH PROGRAM SPONSORED REPORT SERIES

Computing without Revealing: A Cryptographic Approach to eProcurement

5 September 2018

Siva C. Chaduvula
Ashish M. Chaudhari
Jitesh H. Panchal
Mikhail J. Atallah

School of Aeronautics and Astronautics

Purdue University

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-17-1-0009. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Executive Summary

In typical eProcurement processes, sensitive data such as prices, intellectual property, and customer information often flow across enterprise boundaries. Such sharing amplifies the risk of the data breach due to exposure to the potential security flaws of eProcurement partners. Threats of information leakage inhibit enterprises from sharing sensitive data; thus, enterprises cannot take full advantage of the eProcurement process. Existing cryptography-based data sharing protocols impose a high computational burden for maintaining data confidentiality in the procurement process. This additional burden makes existing cryptographic approaches unsuitable for real-time applications. With this motivation, we address the following research question: How can procurers and suppliers securely conduct their business transactions without revealing their confidential information?

The technical approach for addressing the research question consists of developing foundational protocols for secure lightweight computations. The approach enables procurers and suppliers to perform computations while preserving their confidential information. In this report, we show how Computing-without-Revealing (CWR)-based data sharing protocols can be used as building blocks to execute auctions in the procurement process for standard products and innovative technologies. The design of a software embodiment of these protocols as a web-based platform is described. The platform is used to conduct experiments for measuring the performance of the developed protocols against competing techniques. Experimental results corroborate the efficiency of the developed protocols, making them suitable for real-time applications. The application of the protocols is demonstrated for different eProcurement scenarios, including first and second-price auctions for standard products. Pilot laboratory experiments were conducted to understand the behavioral implications of using these protocols in eProcurement scenarios. Results from the pilot experiment are discussed.

Keywords: Secure multi-party computations, eProcurement, Auctions

THIS PAGE INTENTIONALLY LEFT BLANK

PUR-CM-18-228

Acquisition Research Program
sponsored Report Series

Computing without Revealing: A Cryptographic Approach to eProcurement

5 September 2018

Siva C. Chaduvula
Ashish M. Chaudhari
Jitesh H. Panchal
Mikhail J. Atallah

School of Aeronautics and Astronautics

Purdue University

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-17-1-0009. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

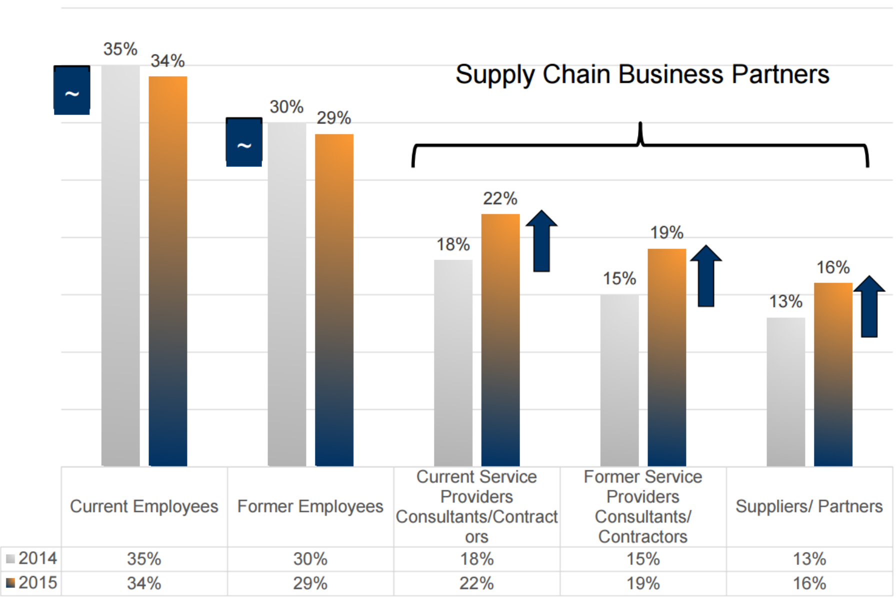
List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Introduction

The design and manufacturing of products, regardless of complexity, involve partnerships with third-party vendors, manufacturers, suppliers, contractors, and other entities outside the organization. The design of a Boeing 777 airplane, for example, involved over 10,000 people external to Boeing. Similarly, Ford Motor Company works with over 1,000 suppliers across the globe. Such partnerships allow organizations to focus on their core expertise, thereby increasing their effectiveness. However, there are also risks associated with sharing confidential information with business partners. In the 2016 acquisition research symposium, it was highlighted that business partners pose a significant malicious threat because they are a part of the information flow (see Figure 1). Therefore, there is a growing need for research and development on technologies that enable business transactions without revealing confidential information of the participants.

Traditionally, business transactions between a procurer and suppliers involve a Trusted Third Party (TTP), say a cloud service provider. The procurer and suppliers send their confidential information to the TTP, who performs the required computation. Although this is easy to implement, the main risk is that rogue employees of the TTP (e.g., the people who maintain and update cloud servers) can learn the confidential information. Additionally, information may be compromised through a break-in by hackers, through a malware or spyware infestation, or even in a completely non-malicious (i.e., accidental) manner. There is also a potential risk that the cloud service provider may, as an organization, decide to betray the users by revealing or secretly using their confidential inputs. A recent report from Ponemon highlighted the impact of internal attacks by insiders/contractors on organizations (see Figure 2). Therefore, it is important to preserve the confidentiality of an organization’s data while engaging with current and especially potential suppliers.



	Current Employees	Former Employees	Current Service Providers Consultants/Contractors	Former Service Providers Consultants/ Contractors	Suppliers/ Partners
▨ 2014	35%	30%	18%	15%	13%
■ 2015	34%	29%	22%	19%	16%

* Adapted from the PwC *The Global State of Information Security® Survey 2016*,

Figure 1. Incidents of data breaches among business partners [Kaestner et al. 2016].

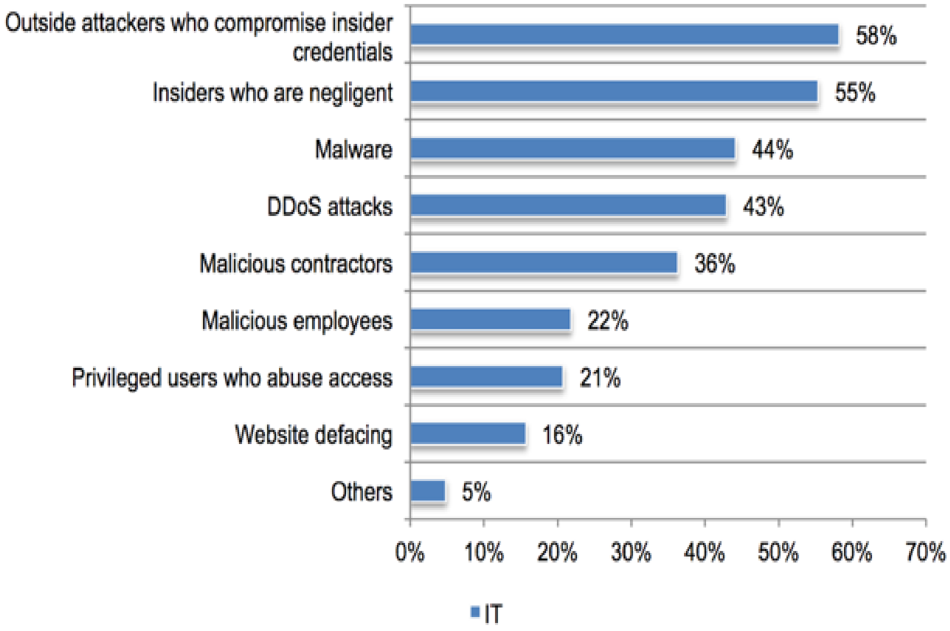


Figure 2. Influence of different security threats faced by organizations [Ponemon 2018]

In a typical eProcurement process, sensitive information related to prices, intellectual property, and customer data often flow across enterprise boundaries. While this data flow between eProcurement partners is important for performing business operations, there exist data security concerns, especially when the data involves intellectual property, trade secrets, etc. Sharing such confidential data amplifies the risk of data breach due to potential security flaws of the partners in the eProcurement process. Such threats discourage enterprises from sharing sensitive data and thus prevent these enterprises from taking full advantage of the eProcurement process.

In this report, we present an approach for addressing this fundamental challenge. The approach enables secure eProcurement of standard products as well as innovative technology. We present the use of cryptographic protocols to execute auction mechanisms within an eProcurement process, where the procurer only learns confidential information related to winning bidders. No confidential information about the losing bidders is revealed to anyone, including the procurer, thereby resulting in truthful revelation and increasing value for all participants involved. This proposed eProcurement process promises economic advantages for a wide variety of private-sector organizations ranging from large electronics manufacturers and automakers to small and medium-sized enterprises specializing in specific technologies.

THIS PAGE INTENTIONALLY LEFT BLANK

Overview of the Approach

Current procurement processes are characterized by incomplete and disaggregated information about (i) the capabilities and cost structure of individual suppliers and (ii) the requirements of the procurers. In a typical eProcurement process, such as a sealed-bid reverse auction as shown in Figure 3, procurement happens in three stages. In Stage 1, the procurer reveals his/her requirements to the suppliers. In Stage 2, suppliers submit their consolidated bids. In Stage 3, the procurer analyzes the submissions and determines the winner by choosing the supplier with the best technology at the lowest bidding price.

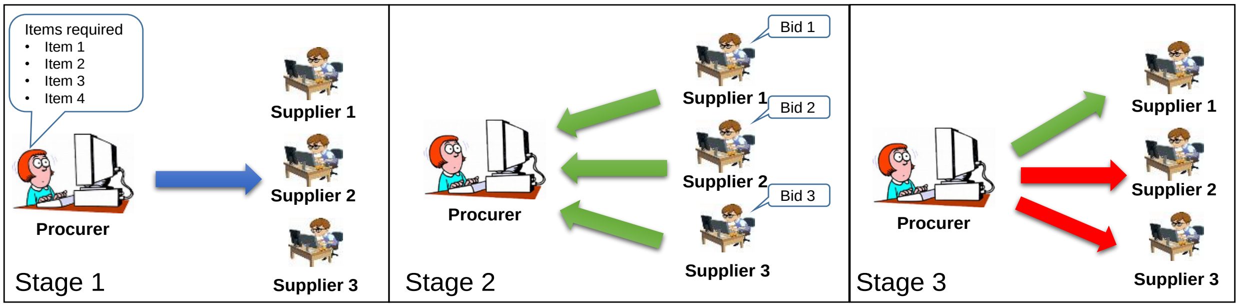


Figure 3. Existing approach for sealed-bid auctions

In such a setting, suppliers would ideally like the procurer to learn their confidential cost information and the details of the proprietary technology only if they win the contract. However, procurers need to determine the quality and suitability of the technology to choose the winner. In addition, procurers may not want to reveal their requirements, especially if the requirements reveal their competitive advantage. This reluctance to reveal sensitive information may drive the procurer to settle for inferior solutions, thereby reducing the overall effectiveness of the procurement mechanism. This brings us to the **research question** addressed in this study: *How can procurers and suppliers securely conduct their business transactions without revealing their confidential information?*

Our **central hypothesis** for this project is that the fundamental protocols discussed in Section 3 can be used as building blocks to perform the computations involved in an eProcurement process. Computational results derived using Computing-without-Revealing (CWR) protocols help in reducing information asymmetry while also protecting the sensitive information held by procurers and suppliers. Such an approach enables procurers and suppliers to estimate the challenges and uncertainties involved and thereby help both sides of the eProcurement process make informed decisions. In this section, we describe the roadmap of research activities that allow us to test our central hypothesis.

Procurement processes based on the proposed CWR approach would enjoy the following benefits:

- **No cryptographic key management:** No data is lost if the secret key used for determining the splits is inadvertently lost.
- **Computation time:** The proposed protocols are computationally lightweight, unlike homomorphic encryption and circuit evaluation. Hence, it is possible to perform huge computations with weaker and battery-powered portable devices such as smart phones.
- **No data abuse:** The data is handled by cloud servers, procurer(s), and supplier(s). In our approach, no user will know the actual inputs of their counterparts. Hence, there is no scope for misusing the data. Even if there is a breach in one of the cloud server(s), the data that a hacker can access would only be a share of the actual data.
- **No complex infrastructure required:** Because their confidential information is protected, procurers and suppliers can use cloud services for procurement processes. This has cost advantages in terms of capital expenditure, IT expenses, etc.
- **Overcomes supplier vulnerabilities:** The procurer need not worry about a data breach at the supplier's end as the data breached (if any) at the vendor's end will be only a share of the actual data. Therefore, no meaningful data would be leaked.

A sub-field within cryptography called "secure multi-party computations" (SMC) focuses on enabling multiple parties to jointly process their individual confidential data into useful information while preserving the confidentiality of the data belonging to each party. Existing cryptographic practices to perform computations securely can be classified into two broad categories:

1) **No Need of a Third Party:** Cryptographic techniques such as fully homomorphic encryption [Bogetoft et al. 2009], secure circuit evaluation [Ben-

David et al. 2008], and Partial Homomorphic Encryption (PHE) [Paillier, 1999] use encryption-based techniques to hide confidential data. Encrypted data is exchanged between parties and computations are performed on the exchanged encrypted data. Such computations impose a very high computational burden and the times reported using these techniques are much longer than in the case of the traditional TTP approach, which makes them ill-suited for use in practical scenarios.

2) **No Need to Reveal to the Third Party:** On the other hand, using secret sharing techniques is a way to distribute a secret (or confidential data) among a group of parties, where every party is allocated a share of the secret. This secret can be reconstructed only when a sufficient number of shares are combined. Individual shares do not infer anything about the whole secret.

Secret sharing approaches are comparatively faster than encryption-based secure computation techniques. However, the computational burden of these techniques is still considerably high. The cryptographic approach proposed in this study reduces the computational burden, which makes it easier to adapt. Moreover, as the proposed approach is based on general arithmetic primitives, it is well-suited for quickly building secure collaborative computing platforms for new procurement scenarios or for variants of the current state of practice, such as volume-based pricing, which is not handled in previous work.

THIS PAGE INTENTIONALLY LEFT BLANK

Details of the Technical Approach

eProcurement involves standard processes such as Request for Proposals (RFPs), auctions, payments, etc. Usually, these processes require inputs from both procurers and suppliers. We present a secure multi-party computation (SMC) technique that allows procurers and suppliers to perform the computations involved in these standard processes without needing to reveal their confidential inputs to anyone. We term our approach of the SMC technique as Computing-without-Revealing (CWR). It builds on the protocols developed by the PIs, which are presented in [Wang et al. 2017]. The approach is based on two key principles [Wang et al. 2013]:

- Adding/multiplying an input with a random number hides the value of the input. If the random number is much larger than the input, it also hides the order of magnitude.
- Adding/multiplying with a large number is orders of magnitude faster than the use of expensive cryptographic techniques such as homomorphic encryption and secure circuit evaluation.

Consider a scenario where the confidential value is 11. We additively split the value into random-looking shares and a participating cloud server sees only one of the random-looking shares. For example, the additive splits of 11 could be 1819 and –1808 (see Figure 4); it could just as well have been 103 and –92 or –19 and 30. These additively split values of 11 are stored in two different cloud servers. We developed protocols for basic arithmetic operations on such additive splits (see [Wang et al. 2017] for details).

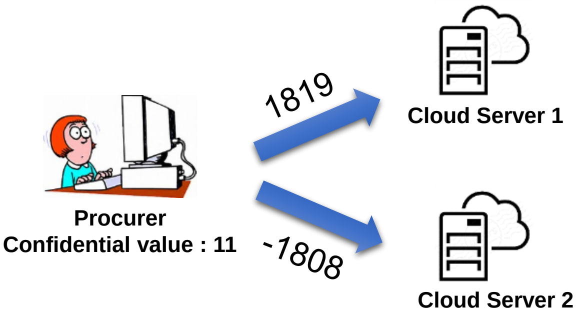


Figure 4. Additive splits

The CWR approach utilizes these splits to perform the desired computation without revealing the input data to anyone. In the next section, we review the existing CWR protocols.

Foundational Computing-without-Revealing (CWR) protocols

CWR protocols enable a procurer (referred to as Alice) and suppliers (referred to as Bob) to use a single external server (referred to as Helper) to perform the computations that are mutually agreed upon between Alice and Bob. The following is the generalized structure of the CWR protocols:

Stage 1- Pre-processing of inputs: The pre-processing of inputs involves two steps:

(a) Split the inputs additively if the inputs from Alice/Bob are not additive splits.

(b) Alice/Bob agree on a morphing function and a distribution from which random numbers are generated. Alice/Bob morph the additive splits using this morphing function and random numbers from the distribution. These morphed additive splits prevent the Helper from learning about Alice/Bob when shared with the Helper.

Stage 2- Run the desired computations securely: Alice/Bob derive the application logic from their mutually agreed computation. Alice/Bob provide the application logic along with the morphed additive splits to the Helper. The application logic involves the sequence of computations that need to be performed on the morphed additive splits. The output derived from running the application logic is additively split. One of the additive splits corresponding to the output is shared with Alice and the other with Bob.

Stage 3- Post processing of outputs: Alice and Bob post-process their additive splits before sharing them with each other. Alice and Bob simultaneously exchange the processed outputs with each other. Alice and Bob independently add their additive splits and learn the actual output of the computation.

Next, we present three existing CWR protocols: CWR-MP, CWR-GT0, and CWR-VIP [Wang et al. 2017]. The first two protocols are focused on performing fundamental operations such as multiplication and greater than zero comparison. These foundational protocols have been utilized to develop higher level protocols including vector operations and matrix operations. We demonstrate this with CWR-VIP, which uses CWR-MP as a building block to securely perform a vector inner product operation.

CWR Multiplication (CWR-MP)

Protocol 1: CWR-MP

Input: Alice's (represented by A) input is *V* and Bob's (represented by B) input is *U*.

Output: A and B receive y_A and y_B respectively, where $y_A + y_B = V*U$.

Stage 1-Pre-processing of inputs:

Step 1: A additively splits her confidential value V into V_A and V_B such that $V = V_A + V_B$. Similarly, B splits U into U_A and U_B.

Step 2: A→B : V_B [Notation means A sends, to B, V_B]

Step 3: B →A : U_A

Step 4: B ←→A : s, H. A and B mutually agree on a seed (s) to generate a series of random numbers. A and B mutually agree on a helper server (H) to help them compute the product of their confidential values. A and B generate random numbers, r_1, r_2, r_3, and r_4, using seed (s). A uses a morphing function $M_A(x) = r_ix+r_{i+1}$ to transform her additive splits (V_A and U_A) into $r_1V_A+r_2$ and $r_3U_A+r_4$, respectively. Similarly, Bob uses morphing function $M_B(x) = r_ix-r_{i+1}$ to transform his additive splits (V_B and U_B) into $r_1V_B-r_2$ and $r_3U_B-r_4$, respectively.

Stage 2- Run desired computations securely:

Step 5: A →H : $r_1V_A+r_2$, $r_1U_A+r_2$

Step 6: B →H : $r_1V_B-r_2$, $r_1U_B-r_2$

Step 7: H: $(r_1V_A+r_2+ r_1V_B-r_2) * (r_3U_A+r_4+r_3U_B-r_4) = r_1 r_3VU=P$. Server splits P into P_A and P_B such that $P = P_A + P_B$.

Stage 3-Post-processing of outputs:

Step 8: H →A : P_A Alice computes $\frac{P_A}{r_1r_3}$

Step 9: H →B : P_B. Bob computes $\frac{P_B}{r_1r_3}$

Correctness

This can be verified by adding the splits held by A and B i.e., $\frac{P_A}{r_1r_3} + \frac{P_B}{r_1r_3} = VU$

Security

The inputs received by H are masked with r_1 and r_3. The product is masked with r_1r_3. Alice receives U_A and cannot determine (U) which is Bob's input. Alice receives P_A from H and cannot determine the product (VU) without P_B . Similarly, Bob receives V_B and cannot determine (V) which is Alice's input. Bob receives P_B from H and cannot determine the product (VU) without P_A.

CWR Greater than Zero (CWR-GT0)

Protocol 2: CWR-GT0

Input: Alice's (represented by A) input is *V* and Bob's (represented by B) input is *U*.

Output: A and B receive b_A and b_B respectively, where $b_A + b_B = 1$ if V>U and 0 else.

Stage 1-Pre-processing of inputs

Step 1: A additively splits her confidential value V into V_A and V_B such that $V = V_A + V_B$. Similarly, B splits U into U_A and U_B.

Step 2: A→B : V_B [Notation means A sends, to B, V_B]

Step 3: B →A : U_A

Step 4: B ←→A : s, H. A and B mutually agree on a seed (s) to generate a series of random numbers. A and B mutually agree on a helper server (H) to help them compute the product of their confidential values. A and B generate random numbers, r_1, r_2, r_3, and r_4, using seed (s). A uses a morphing function $M_A(x) = r_ix+r_{i+1}$ to transform her additive splits (V_A and U_A) into $r_1V_A+r_2$ and $r_3U_A+r_4$, respectively. Similarly, Bob uses morphing function $M_B(x) = r_ix-r_{i+1}$ to transform his additive splits (V_B and U_B) into $r_1V_B-r_2$ and $r_3U_B-r_4$, respectively.

Stage 2- Run desired computations securely

Step 5: A →H : $r_1(V_A- U_A)+r_2$

Step 6: B →H : $r_1(V_B- U_B)-r_2$

Step 7: H: if $r_1(V_A- U_A + V_B- U_B)>0$ *b = 1, else b = 0.* Server splits b into b_A and b_B such that $b = b_A + b_B$.

Stage 3-Post-processing of outputs:

Step 8: H →A : b_A. If $r_1>0$, Alice sets $b_A = b_A$ *else:* $b_A = -b_A$

Step 9: H →B : b_B. if $r_1>0$, Bob sets $b_B = b_B$ *else:* $b_B = 1-b_B$

Correctness

This can be verified by adding the splits held by A and B i.e. $b_A + b_B = 1$ if V>U else 0.

Security

The inputs received by H are masked with r_1 and the output *b* is hidden with the sign of r_1. Alice and Bob receive only a split of their counterpart's input which will not leak any information on the inputs.

CWR-Vector Inner Product (CWR-VIP)

Protocol 3: CWR-VIP

Input: Alice's (represented by A) input is ***V*** and Bob's (represented by B) input is ***U***. Let the vectors **V** and **U** be of length *n*.

Output: A and B receive y_A and y_B, respectively, where $y_A + y_B$ = ***V.U***.

Stage 1-Pre-processing of inputs:

Step 1: A additively splits her confidential value **V** into $\mathbf{V_A}$ and $\mathbf{V_B}$ such that $\mathbf{V = V_A + V_B}$. Similarly, B splits **U** into $\mathbf{U_A}$ and $\mathbf{U_B}$.

Step 2: A→B : $\mathbf{V_B}$ [Notation means A sends, to B, $\mathbf{V_B}$]

Step 3: B →A : $\mathbf{U_A}$

Step 4: B ←→A : s, H. A and B mutually agree on a seed (s) to generate a series of random numbers. A and B mutually agree on a helper server (H) to help them compute the vector inner product of their confidential values. A and B generate *4n* random numbers using seed, s, to morph the additive splits (as described in Step 4 of CWR-MP). A morphs the elements in $\mathbf{V_A}$ followed by $\mathbf{U_A}$ using the morphing function $M_A(x) = r_i x + r_{i+1}$. Similarly, Bob uses morphing function $M_B(x) = r_i x - r_{i+1}$ to transform all the elements of ($\mathbf{V_B}$ and $\mathbf{U_B}$) in the same order.

Stage 2- Run desired computations securely

Step 5: A and B run steps 5-7 of CWR-MP with the help of H for all the elements in $\mathbf{V_A}$. For this, the additive splits from A and B are $(V_{A,i}\ ,U_{A,i})$ and $(V_{B,i}\ ,U_{B,i})$, respectively. H stores the resulting outputs that belong to A and B in the i^{th} position of vector $\mathbf{P_A}$ and $\mathbf{P_B}$**,** respectively.

Stage 3-Post-processing of outputs:

Step 6: H →A : $\mathbf{P_A}$. Alice computes $y_A = \sum_{i=1}^{n} \frac{P_{A,i}}{r_{4i+1} r_{4i+3}}$

Step 7: H →B : $\mathbf{P_B}$ Bob computes $y_B = \sum_{i=1}^{n} \frac{P_{B,i}}{r_{4i+1} r_{4i+3}}$

Correctness

This can be verified by adding the splits held by A and B i.e., $y_A + y_B = V.U$

Security

The inputs received by H are masked with random numbers and the output is hidden within the product of random numbers. Alice and Bob receive only a split of their counterpart's input which will not leak any information on the inputs.

Extension of CWR to eProcurement

We investigated the different types of computations involved in a procurement process and learned that computations such as order statistics are very common in eProcurement processes. Existing CWR protocols that are aimed at computing arithmetic and logical operations can be constructed to perform order statistics, such as minimum. However, they are resource intensive. Therefore, we developed dedicated CWR protocols for computing order statistics such as minimum.

New CWR protocols

In this section, we present two dedicated CWR protocols (Protocol 4 and Protocol 5) for determining the minimum of confidential inputs in constant time. Protocol 4 and Protocol 5 differ in terms of information that is known to the participants of the procurement process. Protocol 4 assumes that the participants are aware of the confidential inputs whereas Protocol 5 assumes that the participants of the procurement process know only an additive split of the confidential inputs. Note that Protocol 4 and Protocol 5 can be used for determining the maximum of the confidential inputs upon changing the sign of the inputs.

CWR-Minimum

Protocol 4: CWR-Min

Input: Alice's (represented by A) input is *a* and Bob's (represented by B) input is *b*.

Output: A and B receive y_A and y_B, respectively, where $y_A + y_B = \min(a,b)$.

Stage 1-Pre-processing of inputs:

Step 1: Both A and B agree on the range and precision for their confidential values. For simplicity, let the range be denoted by [0, R] and precision be denoted by *p*.

Step 2: A transforms her confidential input *a* into a vector **V** as follows:

$$\prod_{i=1}^{n=R/p} v_i = \begin{cases} 1, if\ a > ip \\ 0, otherwise \end{cases}$$

Similarly, B transforms his confidential input *b* into a vector **U**.

Step 3: A additively splits the elements of her confidential vector into $\mathbf{V_A}$ and $\mathbf{V_B}$ such that $\mathbf{V = V_A + V_B}$. Similarly, B splits **U** into $\mathbf{U_A}$ and $\mathbf{U_B}$.

Step 4: A→B : $\mathbf{V_B}$ [Notation means A sends, to B, $\mathbf{V_B}$]

Step 5: B →A : $\mathbf{U_A}$

Stage 2- Run desired computations securely

Step 6: A and B with the help of helper H perform Step 4 and Step 5 of the CWR-VIP to compute the scalar product using their additive splits ($\mathbf{V_A, U_A}$) and ($\mathbf{V_B, U_B}$), respectively. At the end of this protocol, H stores the resulting P_A and P_B in the ith position of vector $\mathbf{P_A}$ and $\mathbf{P_B}$.

Stage 3-Post-processing of outputs

Step 7: H →A : $\mathbf{P_A}$. Alice computes similar to Step 6 of the CWR-VIP and multiplies with *p* to determine y_A

Step 8: H →B : $\mathbf{P_B}$. Bob computes similar to Step 7 of CWR-VIP and multiplies with *p* to determine y_B

Correctness

Upon converting a and b into vectors **V** and **U**, the one that is minimum between a and b will contain a greater number of zeroes. A scalar product between **V** and **U** would result in the minimum of a and b.

Security

H cannot learn the inputs as they are masked with random numbers. In addition, H cannot learn the output, which is hidden within the product of random numbers. Alice and Bob receive only a split of their counterpart's input, which cannot be used to leak any information on the inputs.

CWR-Min Splits

Protocol 5: CWR-Min Splits

Input: Alice's (represented by A) input is (a_A, b_A) and Bob's (represented by B) input is (a_B, b_B).

Output: A and B receive y_A and y_B, respectively, where $y_A + y_B = \min(a_A + a_B , b_A + b_B)$.

Stage 1-Pre-processing of inputs

Step 1: Both A and B agree on the range and precision for their confidential values. For simplicity, let the range be denoted by [0, R] and precision be denoted by p.

Step 2: A←→B: $\boldsymbol{X}_A$, $\boldsymbol{X}_B$. A and B mutually split the vector $\boldsymbol{X} = [0,\ \mathrm{p},\ 2\mathrm{p}\ \ldots, \mathrm{ip}, \ldots, \frac{\mathrm{R}}{p}]$ into $\boldsymbol{X_A}\ and\ \boldsymbol{X}_B$ such that $\boldsymbol{X} = \boldsymbol{X_A} + \boldsymbol{X}_B$.

Stage 2- Run desired computations securely

Step 3: A and B with the help of an external server as Helper (H) perform Steps 5-7 of the CWR-GT0 protocol using (a_A, a_B) and additive splits of ($\boldsymbol{X}_A$, $\boldsymbol{X}_B$) as inputs and determine $\mathbf{V_A}$ and $\mathbf{V_B}$.

Step 4: Similarly, A and B with the help of H perform Steps 5-7 of the CWR-GT0 protocol using (b_A ,b_B) and additive splits of ($\boldsymbol{X}_A$, $\boldsymbol{X}_B$) as inputs and determine vector $\mathbf{U_A}$ and $\mathbf{U_B}$.

Step 5: A and B with the help of H compute the scalar product using their additive splits ($\mathbf{V_A}$**,** $\mathbf{U_A}$) and ($\mathbf{V_B}$**,** $\mathbf{U_B}$), respectively, via Steps 5-7 of the CWR-MP protocol. At the end of this protocol, A and B receive y_A and y_B. Further, H stores the resulting P_A and P_B in the ith position of vectors $\mathbf{P_A}$ and $\mathbf{P_B}$.

Stage 3-Post-processing of outputs

Step 6: H →A : $\mathbf{P_A}$ Alice computes similar to Step 6 of the CWR-VIP and multiplies with *p* to determine y_A.

Step 7: H →B : $\mathbf{P_B}$ Bob computes similar to Step 7 of the CWR-VIP and multiplies with *p* to determine y_B.

Correctness

Upon converting a and b into vectors **V** and **U**, the one that is minimum among a and b will contain a greater number of zeroes. A scalar product between **V** and **U** would result in the minimum of a and b.

Security

H cannot learn the inputs as they are masked by random numbers. In addition, H cannot learn the output, as it is hidden within the product of random numbers. Alice and Bob receive only a split of their counterpart's input, which will not leak any information on the inputs.

Our ultimate goal for this project is to test the feasibility of a procurement platform that enables participants of the procurement process to execute the computations involved without having to know the details of how the CWR technology operates. This allows procurers and suppliers to focus on designing a well-suited eProcurement process for their business needs. The research tasks performed to achieve this goal are as follows:

1: Extension of CWR to auctions for standard products

2: Extension of CWR to the procurement of innovative technology

3: Performance analysis of CWR

4: Influence of CWR on human behavior

In the following sections, we describe these research tasks in detail.

Extension of CWR to auctions for standard products

The objective in this research task was to extend CWR to eProcurement for standard products or commercial-off-the-shelf items. In other words, the quality of these types of products is established.

While there are many ways to perform auctions within an eProcurement process for standard products, in what follows, we use reverse sealed-bid auctions to illustrate how CWR protocols can be used as building blocks to perform the computations involved (as shown in Figure 4). Note that the CWR-protocols can be constructed to perform the computations involved in any auction mechanism, but to simplify the discussion, we focus on the first price reverse sealed-bid auction. The computation involved in such auctions is the identification of a supplier with the minimum consolidated bid for all the items listed by the procurer. The procurer and suppliers mutually agree on three external servers (for example, cloud servers $\alpha, \beta, \text{and } \gamma$). The procurer provides unique IDs to all the suppliers. Suppliers share the additive splits corresponding to their confidential information (i.e., consolidated bids) along with their IDs with cloud server α and cloud server β. Cloud server α (as Alice) and cloud server β (as Bob), together with cloud server γ (as Helper), deploy protocol 5. After protocol 5 ends, the cloud servers $\alpha \text{ and } \beta$ share the additive splits obtained with the procurer. By adding these additive splits, the procurer finds the supplier with the minimum consolidated bid and the value of the consolidated bid.

This extension of CWR to eProcurement enables procurers and suppliers to perform procurement transactions without needing to reveal their confidential information to anyone. This allows procurers to design auction mechanisms that can help them overcome inefficiencies in existing auction mechanisms. For example, an auction mechanism built using CWR can identify the supplier with the best price (i.e., “cherry pick” the suppliers) for each and every item. Such an auction mechanism has great potential to reduce procurement costs, as the procurer gets the best possible price for every item. This will appeal to suppliers as well because their individual item prices are not revealed to anyone, including to the procurer. In this section, we present a CWR first price reverse auction that enables the procurer to select the supplier who provides the greatest “bang for their buck” for each individual item and thereby overcome this inefficiency.

CWR first price reverse auction

In a CWR first price reverse auction, a single procurer (say, the DoD) can “cherry pick” the best supplier among the suppliers (DoD contractors) for each and every item. The CWR first price reverse auction is listed in Protocol 5.

The CWR first price reverse auction enables the procurer to learn only the payments that need to be made to each individual supplier and the items provided by each supplier. Throughout the protocol, the procurer cannot learn the supplier’s individual item prices. Similarly, the supplier cannot learn the quantity desired by the procurer before the auction. The novelty in this protocol is that the external servers (cloud servers $\alpha, \beta, \text{and } \gamma$) on which the CWR protocols are run do not know the auction’s context (item names, etc.) as they receive morphed additive splits. Therefore, the external servers learn nothing about the procurer’s/supplier’s confidential information. Note that this protocol is designed to choose the supplier based on a single attribute of the product (price). This protocol can be extended to multiple attributes with the appropriate weights.

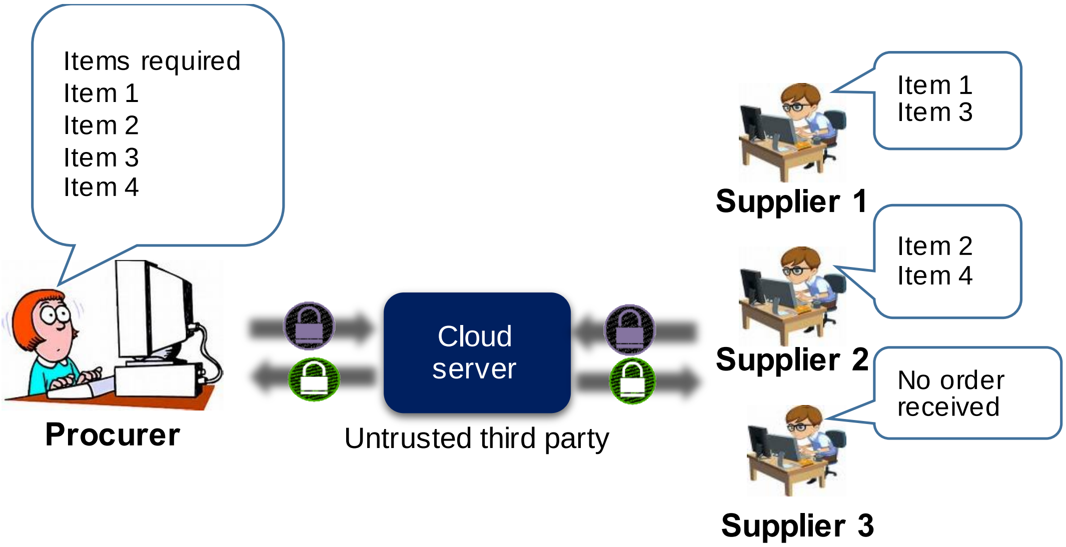


Figure 5. CWR-first price reverse auction

Implementation details

Below are some of the details for implementing the CWR first price reverse auction:

1. **Secure Channels**: It is important to understand that information exchanges that occur between parties within the CWR auction should use secure channels, such as HTTPS.
2. **Cross Accounts:** The ownership of the cloud server account is one of the concerns while deploying CWR. Existing cloud servers, such as Amazon Web Services (AWS), offer features such as cross accounts through which a procurer and suppliers can examine what algorithms are being run on their data splits. Please refer the webpage in this link (https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html) for more details.
3. **Tie Breaks**: There is a possibility that the item prices of suppliers may be the same. In such scenarios, the procurer can break such ties in many ways, including randomly picking a supplier from the suppliers with the same item price. How such ties are handled is made public to all participants prior to the auction.
4. **Single Item Winner:** In some scenarios, a supplier may win only one item. This can reveal the item price to the procurer when he/she makes payments. In such scenarios, the corresponding supplier is informed and the supplier may choose to participate/quit the procurement process.

Protocol 6: CWR-first price reverse auction

Input: Procurer provides the list of items (denoted by $\boldsymbol{I}$) and their respective quantities (denoted by $\boldsymbol{q} = [q_1, \ldots, q_N]$). Suppliers ($S_1, \ldots, S_K$) provide their item prices for the items in the list $\boldsymbol{I}$. Supplier S_k item price list is denoted by $\boldsymbol{p_k} = [p_{k1}, \ldots, p_{kN}]$.

Output: Procurer determines the items won (represented by $\boldsymbol{w_k}$) by each supplier S_k and payment (represented by a_k).

Stage 1-Pre-processing of inputs

Step 1: The procurer and suppliers mutually identify cloud servers (α and β) as their surrogates to execute procurement using CWR. The procurer splits their sensitive information $\boldsymbol{q}$ into $\boldsymbol{q}_\alpha$ and $\boldsymbol{q}_\beta$ such that $\boldsymbol{q} = \boldsymbol{q}_\alpha + \boldsymbol{q}_\beta$ and shares them with the cloud servers α and β, respectively. Similarly, the suppliers split their individual item price list and share them with the cloud servers α and β, respectively.

Step 2: Cloud servers (α and β) mutually agree upon morphing functions (M_α, M_β) and a seed to generate the random numbers that are used in these morphing functions. These agreements can be derived using session number, auction ID etc. Further, the cloud servers (α and β) identify another cloud server (γ) as their helper to perform the desired procurement computations using CWR.

Stage 2- Run desired computations securely

Step 3: Cloud servers ($\alpha, \beta,$ and γ) execute Steps 3-7 of CWR-Min followed by Steps 5-9 of CWR-MP for all the items listed in $\boldsymbol{I}$. Cloud servers (α and β) keep track of the splits corresponding to the information on whether a supplier S_k won/lost the items ($\boldsymbol{w_{\alpha k}} = [w_{\alpha 1}, \ldots, w_{\alpha N}]$, $\boldsymbol{w_{\beta k}}$) and the splits corresponding to the payments that are to be made to the supplier S_k ($a_{\alpha k}$, $a_{\beta k}$). The vector ($\boldsymbol{w_k} = \boldsymbol{w_{\alpha k}} + \boldsymbol{w_{\beta k}}$) has 1s against the items that are won and 0s against all the items lost by the supplier S_k.

Step 4: By the end of Step 3, cloud server α has ($\boldsymbol{a_\alpha} = [a_{\alpha 1}, \ldots, a_{\alpha k}]$, $\boldsymbol{W_\alpha} = [\boldsymbol{w_{\alpha 1}}, \ldots, \boldsymbol{w_{\alpha K}}]$) and cloud server β has ($\boldsymbol{a_\beta}$, $\boldsymbol{W_\beta}$). Both cloud servers (α and β) share their splits with the procurer. The procurer adds ($\boldsymbol{a_\alpha}$, $\boldsymbol{a_\beta}$) to determine $\boldsymbol{a} = [a_1, \ldots, a_K]$ where a_k refers to the money that the procurer owes the supplier S_k. Similarly, the procurer adds ($\boldsymbol{W_\alpha}$, $\boldsymbol{W_\beta}$) to determine $\boldsymbol{W} = [\boldsymbol{w_1}, \ldots, \boldsymbol{w_K}]$.

Stage 3-Post-processing of outputs

Step 5: Procurers provide the payment a_K and items won (represented by $\boldsymbol{w_k}$) to S_k. Supplier S_k verifies the payment a_K against the item prices that he/she won.

Correctness

The correctness is derived from the correctness of CWR-min and CWR-MP.

Security

Procurer knows **q, a,** and ***W***. With this information, the procurer cannot infer the suppliers' item prices. Similarly, suppliers receive $\mathbf{w_k}$ and the items they need to provide to the procurer. Additionally, the suppliers cannot infer each other's private information such as item price. All the external servers $(\alpha, \beta, \text{and } \gamma)$ receive only one of the additive splits. However, these external servers learn the number of suppliers participating in the auction. This could be avoided by using different external servers for the computations.

In the next section, we describe how CWR-protocols can be used while procuring innovative technology.

THIS PAGE INTENTIONALLY LEFT BLANK

Extension of CWR to procurement of innovative technology

Procuring standard products are usually driven by price. However, while procuring innovative goods and services such as an aircraft, in addition to price, the quality of fit and performance characteristics of the supplied product should be evaluated to determine the winner. Usually, these products are also associated with confidential information such as intellectual property. This makes the procurement process for a non-standard product or development products complex.

Information asymmetry that exists between the procurer and suppliers (say, DoD contractors), including product characteristics such as price and quality, inhibits the procurer and suppliers from fully understanding the relative importance between dimensions of quality and price. In this section, we describe how CWR protocols can be used in a procurement process so that the procurer and suppliers can reduce information asymmetry without any need to reveal their individual private information.

Coughlan et al. [2008] proposed an iterated information aggregation auction (I^2A^2) for procuring products where the procurer does not have complete information on the dimensions of the product's quality and their relative importance. Figure 6 shows the different stages in the I^2A^2 mechanism. This mechanism has two auction rounds (Stage 2 and Stage 5). In Stage 2, the procurer collects information on the product's quality from all the suppliers. Using this, the procurer estimates the relative importance of the quality dimensions in Stage 3. In Stage 4, the procurer eliminates suppliers with least-value bids and in Stage 5, the remaining suppliers submit their final bid. In Stage 6, the procurer chooses the supplier with the highest value.

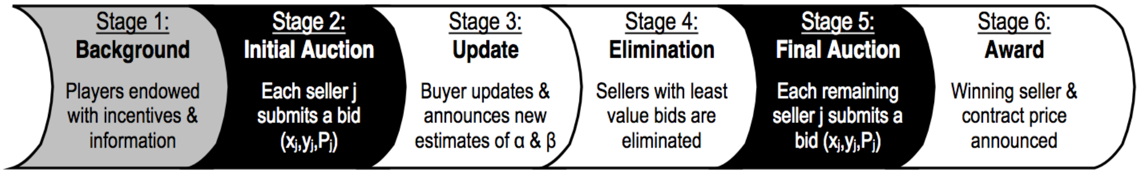

Stage 1:
Background
Players endowed with incentives & information
Stage 2:
Initial Auction
Each seller j submits a bid (x_j,y_j,P_j)
Stage 3:
Update
Buyer updates & announces new estimates of α & β
Stage 4:
Elimination
Sellers with least value bids are eliminated
Stage 5:
Final Auction
Each remaining seller j submits a bid (x_j,y_j,P_j)
Stage 6:
Award
Winning seller & contract price announced

Figure 6. Iterated information aggregation auction (I2A2) mechanism [Coughlan et al. [2008]]

The I^2A^2 mechanism induces suppliers to truthfully reveal information about the procurer value and supplier cost. However, this mechanism favors the procurer alone. In the I^2A^2 mechanism, the suppliers lose ownership of their confidential information and have no control on how their information submitted in Stage 2 can be used by the procurer, apart from drawing an estimate of the relative importance of different dimensions of quality. In addition to this, the disclosed information is prone to risks such as information leak by a disgruntled employee in the procurer's organization. Such risks need to be borne by suppliers without any guarantee of being awarded the contract. We address these concerns by performing the computations involved in Stages 2-5 of this procurement process using CWR protocols.

CWR-I^2A^2 is a procurement process in which the procurer and suppliers mutually pre-agree on computations, including elimination criteria, and suppliers need not reveal any kind of confidential information until they are awarded the contract. CWR-I^2A^2 aims at deriving the aggregates while maintaining the privacy and the diffuse nature of information held with suppliers.

CWR-I^{2}A^2

Protocol 7: CWR-I^{2}A^2

Input: Procurer provides a list of dimensions for quality (denoted by ***D*** = *[d$_1$…,d$_N$]*) and his/her relative importance of the quality dimensions (denoted by ***U*** = *[u$_1$…,u$_N$]*). Suppliers (denoted by S$_K$ where k ∈ $[1, k]$) provide their item price (denoted by *P$_k$)* along with the weights (denoted by ***W$_k$*** = *[w$_{1,k}$…,w$_{N,k}$]*) against the dimensions of quality listed in ***D***. Before the auction process begins, procurer discloses their elimination criteria used in Stage 4. Below is an example of an elimination criteria: $\sum_{k=1}^{K} \mathrm{b}_{i,k} > \frac{K}{2}$ where, $\mathrm{b}_{i,k}$ is given by

$$\mathrm{b}_{i,k} = \begin{cases} 1, & \dfrac{\sum_{k=1}^{K} W_{i,k}}{K} - W_{i,k} < 0 \\ 0, & \dfrac{\sum_{k=1}^{K} W_{i,k}}{K} - W_{i,k} \geq 0 \end{cases}$$

Output: Procurer identifies the supplier S$_K$ who provides the highest value (P'$_K$-**V$_K$.D**).

Stage 1-Pre-processing of inputs

Step 1: Procurer and suppliers mutually identify cloud servers (α and β) as their surrogates to execute procurement using CWR. Procurer splits their sensitive information (***D**, **U**)* into *(**D**$_α$,**U**$_α$*) and (***D**$_β$,**U**$_β$)* such that ***D*** = ***D**$_α$* + ***D**$_β$* and ***U*** = ***U**$_α$* + ***U**$_β$.* Procurer shares *(**D**$_α$,**U**$_α$*) and (***D**$_β$,**U**$_β$)* with cloud servers α and β*,* respectively. Similarly, each supplier (S$_k$) splits their individual item prices *P$_k$* and weights ***W$_k$*** *into (P*$_{αk}$, ***W***$_{αk}$) and (*P*$_{βk}$, ***W***$_{βk}$) and shares them with cloud servers α and β*,* respectively.

Step 2: Cloud servers (α and β) mutually agree upon a seed to generate the required random numbers to morph the additive splits. This seed can be derived from session number etc. Further, the cloud servers α and β identify another cloud server (γ) as their helper to perform the desired procurement computations using CWR.

Stage 2- Run desired computations securely

Step 3: Cloud server α and cloud server β internally compute the values of $\left(\frac{\sum_{k=1}^{K} W_{\alpha i,k}}{K} - W_{\alpha i,k}, \frac{\sum_{k=1}^{K} W_{\beta i,k}}{K} - W_{\beta i,k}\right)$ and use cloud server (γ) to execute CWR-GT0. By the end of this protocol, cloud servers (α and β) obtain the additive splits corresponding to $\mathrm{b}_{i,k}$. This step is repeated for all the quality dimensions and suppliers. Cloud servers (α and β) independently send the additive splits corresponding to $\mathrm{b}_{i,k}$values to the procurer, who determines the eligible suppliers by computing $\sum_{k=1}^{K} \mathrm{b}_{i,k} > \frac{K}{2}$ and informs the suppliers about the outcome.

Step 4: The eligible suppliers reconsider their item prices *P$_k$* and weights ***W$_k$*** and revise their item price *P'$_k$* and weights ***W'$_k$***, if needed. The eligible suppliers submit their additively split price *P'$_k$* and weights ***W'$_k$*** *into (P'*$_{αk}$, ***W'***$_{αk}$) and (*P'*$_{βk}$, ***W'***$_{βk}$). Cloud servers ($\alpha, \beta,$ and γ) together perform a CWR-first price reverse auction. By the end of this step, procurer receives (**a$_α$, a$_β$**) from cloud servers

α and β, respectively. Procurer adds (**a_α, a_β**) to determine ***a*** = *[a_1, …, a_K]* where a_k refers to the money that the procurer owes the supplier S_k. Similarly, procurer adds ***(W_α, W_β)*** to determine ***W*** = [***w_1***, …, ***w_K***].

Stage 3-Post-processing of outputs

Step 5: Procurer provides the payment a_K and items won (represented by ***w_k***) to S_k. Supplier S_k verifies the payment a_K against the item prices that he/she won.

Correctness

The correctness is derived from CWR-first price reverse auction, CWR-min, and CWR-MP.

Security

Procurer knows **q**, **a**, and ***W***. With this information, procurer cannot infer suppliers' item prices. Similarly, suppliers receive **w_k** and the items they need to provide to the procurer. The suppliers cannot infer other suppliers' private information, such as item price. All the external servers ($\alpha, \beta,$ and γ) receive only one of the additive splits of the confidential information. With this, external servers learn nothing.

In the next section, we compare the performance of CWR-based computing techniques with competing secure computing techniques.

Performance Analysis of CWR

We developed a test-bench to run and compare different secure computing techniques such as partial homomorphic encryption and secret sharing, as discussed in Section 2. In what follows, we describe the test bed developed as part of this project to compare our approach (CWR) with the existing cryptographic approaches.

Test-Bed Setup

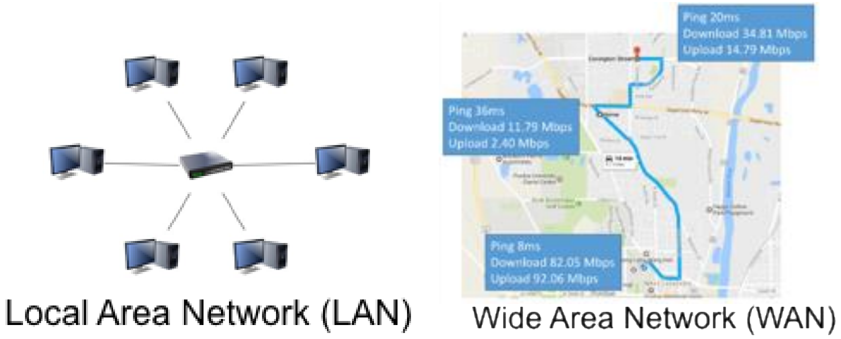


Figure 7. Experimental Setup

We conducted experiments in two different settings. The first set of experiments was conducted when all the procurers and suppliers were connected to the same network (i.e., local area network or LAN). The second set of experiments was conducted when the procurers and suppliers were connected to different networks (i.e., wide area network or WAN). Note that the computation speed of all the approaches reduces with WAN. This is mainly attributed to the network latency.

We identified computational time and bandwidth with respect to the amount of data that needs to be transferred between the procurer/suppliers as the key performance indicators (KPIs). The computational time is measured using a python module named “time” and the bandwidth is measured using an open source packet analyzer (Wireshark). We compared CWR protocols with competing secure computing techniques using these KPIs.

CWR-VIP:

We chose the inner product as the computation to compare the performance of the proposed approach (CWR) against the existing approaches. This computation was chosen as it is commonly used to multiply the vector of quantity with the vector of item prices for the listed items within a procurement process.

We found that the proposed approach (CWR) is at least 10 times faster than the best existing approach (refer Table 1) using LAN. We found that our approach is about 7 times faster than the best existing approach (refer Table 2) using WAN. We realized that the cost of security (computational burden to maintain the confidentiality) in procurement activities is high (about 6-7 times) compared to open sharing, where procurement data is revealed to every participant. One of the reasons for this additional burden is because of the requirement of performing every computation using CWR protocols. In order to reduce this computational burden, we developed a standalone CWR-Min protocol for another commonly occurring operation: minimum.

Table 1. Protocol execution time while using LAN (in seconds)

Vector length	0-server (PHE)	3-servers (Previous best)	1-server (CWR-VIP)
10	14.6	4.1	0.35
100	135.5	37.4	2.88
1000	1738.4	378	27.5
10000	>3600	4031	264.7

Table 2. Protocol execution time while using WAN (in seconds)

Vector length	0-server (PHE)	3-servers (Previous best)	1-server (CWR-VIP)
10	16.5	5.58	0.68
100	235	47.3	6.9
1000	>3600	486.3	74.7
10000	>3600	5567	742.6

In network communication, the amount of data (bandwidth) being exchanged between parties is another important performance indicator. In our comparative study, we found that our approach requires 3 times less bandwidth (refer Table 3). These results indicate that our approach can be deployed in real-time applications and can be supported by devices with limited battery power.

Table 3. Comparison of bandwidth use (in KB)

Vector length	0-server (PHE)	3-servers (Previous best)	1-server (CWR-VIP)
10	6.5	3.4	1.18
100	61.8	33.8	10.6
1000	614.2	342.7	105.9
10000	>5000	3425.3	1053.7

CWR-first price reverse auction

We developed the software embodiment of the CWR-first price reverse auction (described in Protocol 6) and used it as an auction mechanism in a procurement process. We used the values shown in Table 4 to simulate the auction mechanism. In what follows, we describe the outcomes of a traditional sealed bid auction and compare these outcomes with those obtained using CWR-first price reverse auction.

In a traditional sealed-bid auction, the procurer reveals the desired quantity. The suppliers submit their respective sealed bids ($330, $322, $316) to the procurer, who selects the minimum bid ($316) in first price auction and receives the items from Supplier 3. Throughout the auction process, suppliers hide their item prices in the form of sealed bids. However, from Table 4, we learn that Supplier 3 does not provide the best prices for each individual item.

Table 4. Item prices and quantities used for simulation studies

Item Name	Procurer (Quantity)	Supplier 1 (Item price)	Supplier 2 (Item price)	Supplier 3 (Item price)
A	12	$11	$9	$10
B	8	$6.5	$8	$7
C	7	$8	$6	$6.5
D	9	$10	$12	$10.5

Figure 8 shows a picture of the demo of this CWR-first price reverse auction, developed as part of this project. In this demo, one surface pro computer was used as the procurer and three other surface pros were used as the suppliers to simulate a reverse auction. All the surface pros were connected with each other using 2 Mbps (upload/download speed) LAN. The procurer and suppliers mutually agree on three external servers ($\alpha, \beta, \text{and}\ \gamma$) which are used to run the CWR first price reverse auction. A computer is used to run these three external servers and this computer is also connected to all the surface pros using the same LAN.

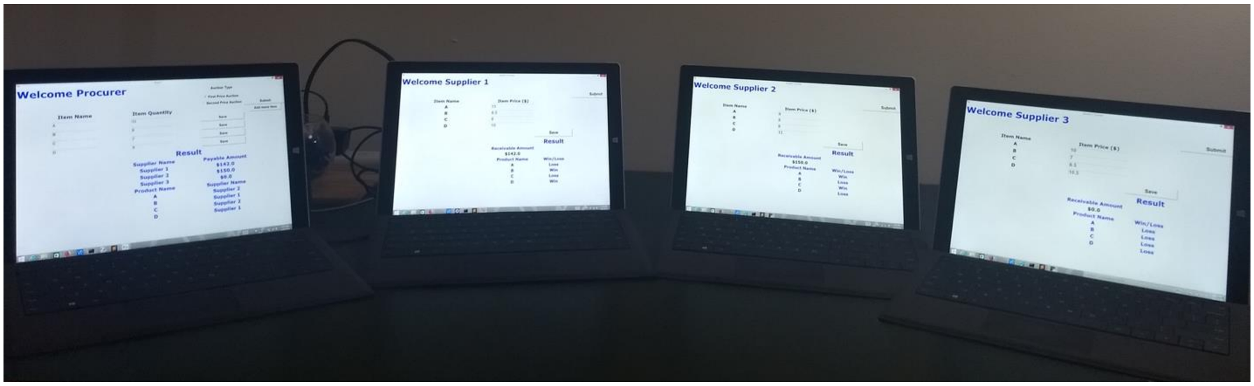
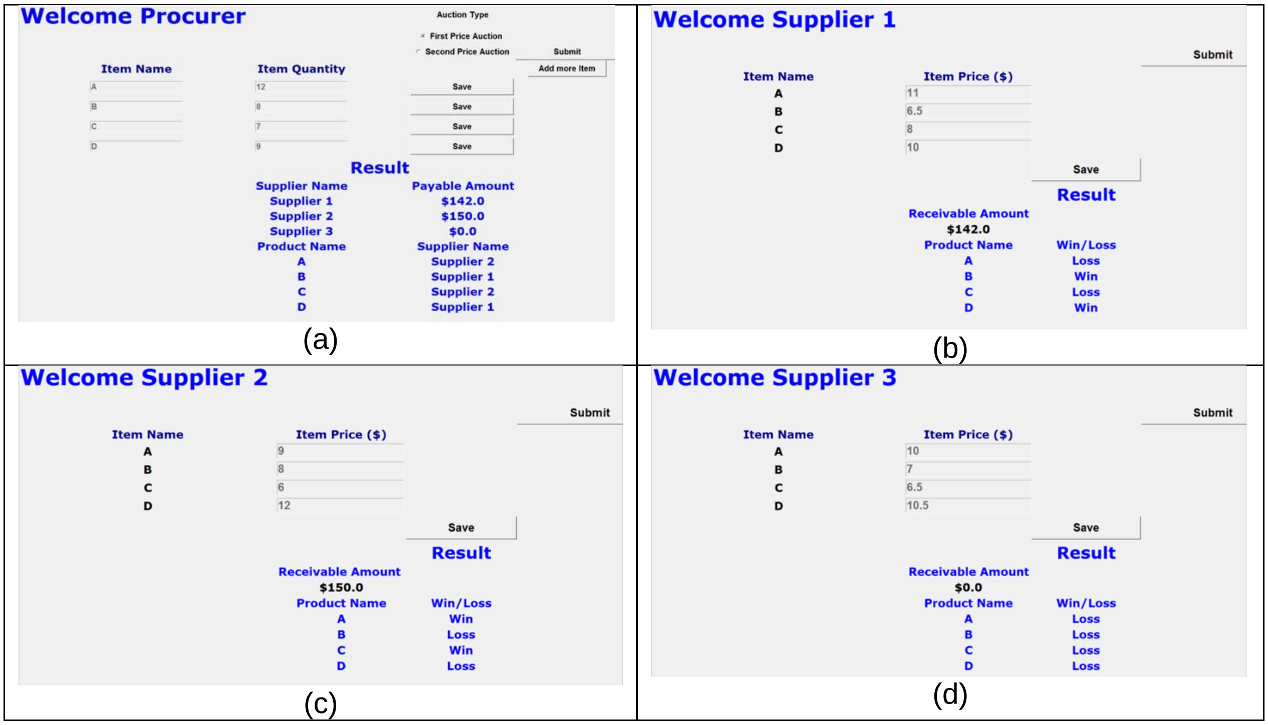


Figure 8. Demo of a CWR-first price reverse auction

Deploying the CWR-first price reverse auction enables the procurer to enter item names and their respective quantities. Only the item names are provided to all the suppliers. Suppliers enter their respective confidential item prices (as listed in Table 4). As described in Protocol 6, the confidential information (item quantities and prices) is split additively and shared with the external servers ($\alpha\ \text{and}\ \beta$). These external servers along with the help of another external server (γ) execute the computations involved in the auction. By the end of these computations, the procurer learns that items (A, C) and (B, D) will be provided by Supplier 2 and Supplier 1, respectively. The procurer also learns the amounts that should be paid to Supplier 1 and Supplier 2. The suppliers also receive information on the items they won/lost and their receivable payout amounts from the procurer. Figure 9 shows the screenshots of the procurer and suppliers at the end of the auction process. Note that throughout the procurement process, suppliers need not disclose their individual item prices to anyone.

This CWR-first price reverse auction enables procurers to select the suppliers who provide the best price for each individual item. Such selection enables the procurer to reduce procurement costs. For instance, using the values listed in Table 4, CWR-first price reverse auction enables the procurer to procure all the desired items for $292 instead of $316 (from traditional sealed-bid auctions). We believe that this form of cherry-picking enables the procurer to increase competition among suppliers and thereby achieve efficient solutions.



Welcome Procurer

Auction Type

● First Price Auction
○ Second Price Auction

Item Name	Item Quantity	
A	12	Save
B	8	Save
C	7	Save
D	9	Save

Result

Supplier Name	Payable Amount
Supplier 1	$142.0
Supplier 2	$150.0
Supplier 3	$0.0

Product Name	Supplier Name
A	Supplier 2
B	Supplier 1
C	Supplier 2
D	Supplier 1

(a)

Welcome Supplier 1

Item Name	Item Price ($)
A	11
B	6.5
C	8
D	10

Result

Receivable Amount
$142.0

Product Name	Win/Loss
A	Loss
B	Win
C	Loss
D	Win

(b)

Welcome Supplier 2

Item Name	Item Price ($)
A	9
B	8
C	6
D	12

Result

Receivable Amount
$150.0

Product Name	Win/Loss
A	Win
B	Loss
C	Win
D	Loss

(c)

Welcome Supplier 3

Item Name	Item Price ($)
A	10
B	7
C	6.5
D	10.5

Result

Receivable Amount
$0.0

Product Name	Win/Loss
A	Loss
B	Loss
C	Loss
D	Loss

(d)

Figure 9. A screenshot of the procurer's screen is shown in Figure 9 (a). Screenshots of the suppliers' screens are shown in Figure 9 (b)-(d).

We extended the functionality of this software embodiment to handle second-price reverse auctions by modifying the calculation of payments in Step 3 of Protocol 6. We tested the scalability of the proposed CWR-first price reverse auction by running for different numbers of items procured by the procurer. The resulting computational time and bandwidth use are reported in Table 5 and 6, respectively. These results indicate that CWR-first price reverse auction is a computationally efficient and secure technique that can be deployed in real-time settings.

Table 5. Comparison of bandwidth use (KB)

Number of items	(CWR: First Price)	(CWR: Second Price)
4	7.8	7.2
8	8.2	8.7
16	9.2	9.5
32	15.3	12.58
64	18.95	18.41

Table 6. Comparison of average computational time (in seconds)

Number of items	(CWR: First Price)	(CWR: Second Price)
4	0.05	0.06
8	0.11	0.13
16	0.21	0.22
32	0.40	0.42
64	0.76	0.82

The CWR-first price reverse auction is a step towards demonstrating that computations in a procurement process can be performed without needing to reveal any confidential information. We believe that procurers and suppliers can build on this and modify it to make it suitable for more sophisticated computations.

Influence of CWR on human behavior

We focused on the design of eProcurement auctions for standard products as the experimental setup to test the influence of protecting confidential information on the strategic behavior of humans (especially among suppliers). We used graduate students as suppliers and the virtual entity as the procurer for this experimental study, which provides a way to simulate information asymmetry among the suppliers. We conducted a pilot experiment with 10 subjects, all graduate students of the School of Mechanical Engineering at Purdue University.

Experiment Task:

In the experiment, every subject was an independent supplier who participated in 20 different sequential first-price auctions, where the supplier with the least bid won and received the winning bid as a reward. In each auction, subjects competed to sell three products, one after other, in three periods. In every period, a subject competed against a single opponent. The cost of the product was constant for all three periods in an auction but varied across auctions. If a subject won with bid b_i and cost of the product c in a period i, then the supplier's profit was $b_i - c$, and 0 if lost. Subjects were instructed to maximize their profit in every period.

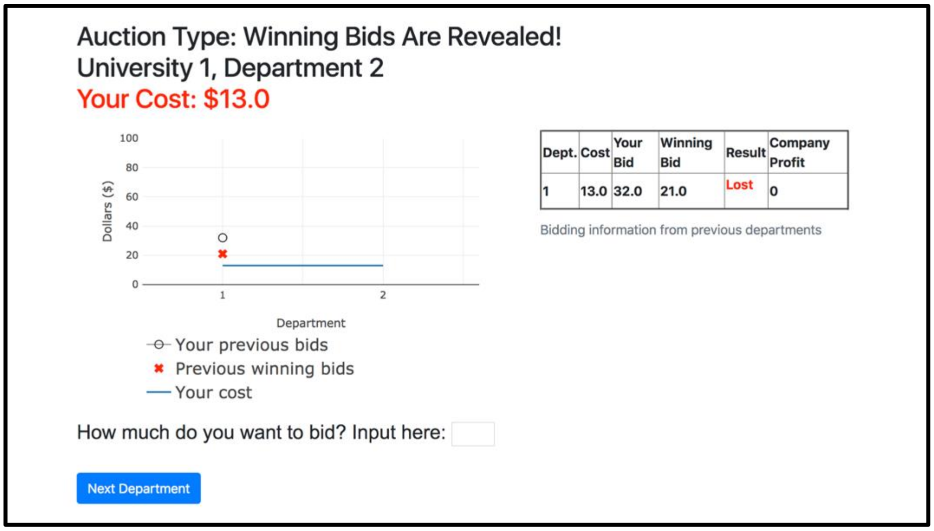
Experimental Treatments:

Four treatments were implemented by varying the policy about winning bid revelation and varying the cost of the product (see Table 7). The two revelation policies were: i) reveal the winning bid (R), and ii) do not reveal the winning bid (NR). The two supplier cost types were: i) low cost of the product (LC), and ii) high cost of the product (HC). Every subject completed five sequential first-price procurement auctions in each of the four treatments. Two subjects, one each of low-cost and high-cost types, were randomly grouped for competing in an auction to avoid collaboration and group strategies.

Figure 10 shows the user interface used for the experimental task. Subjects could see their previous bids and cost at any given time. Winning bids were reported based on the auction type. For representation purposes, the bids were reported both numerically in a table and graphically in a plot.

Table 7. Experiment treatments and the number of auctions in each treatment.

	Winning Bid Revealed (R)	Winning Bid not Revealed (NR)
Low Cost of Product (LC)	Five 3-period sequential auctions	Five 3-period sequential auctions
High Cost of Product (HC)	Five 3-period sequential auctions	Five 3-period sequential auctions



Auction Type: Winning Bids Are Revealed!
University 1, Department 2
Your Cost: $13.0

-o- Your previous bids
✖ Previous winning bids
— Your cost

Dept.	Cost	Your Bid	Winning Bid	Result	Company Profit
1	13.0	32.0	21.0	Lost	0

Bidding information from previous departments

How much do you want to bid? Input here: □

Figure 10. User interface for the experiment developed using oTree [10].

Results (Pilot)

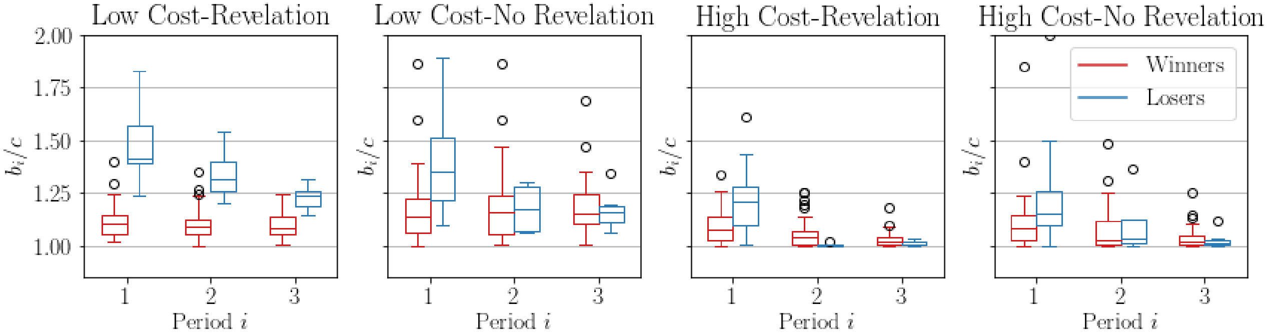


Figure 11. Bid to cost ratio for winners and losers in respective periods.

Figure 11 summarizes the results as subjects' bid to cost ratio ($\frac{b_i}{c}$) for three periods in each of the four treatments. The most common strategy of winners was to increase their bids for the next period, while that of losing subjects was to decrease bids. Given that low-cost type subjects were at an advantage to bid smaller values over high-cost type subjects, the winner group included mainly low-cost type subjects. In treatment R, low-cost type subjects were able to identify the high-cost values in respective auctions. Consequently, the average bid to cost ratio for winning low-cost type subjects was 1.1, which equaled the high-cost to the low-cost ratio.

We observe that inflation of bids relative to the cost of the product by winning subjects is higher in NR treatment (when winning bids are not revealed) than R treatment (when winning bids are revealed). We ran a two-sided t-test between bid inflation by winners of period 2 ($b_3/c - b_2/c$) in R treatment and that in NR treatment. We observe that bid inflation is larger in NR treatment with t-statistic and p-value respectively equal to 2.41 and 0.017. For only low-cost type subjects, t-statistic and p-value values of the same two-sided t-test are 2.88 and 0.005 respectively.

With two players and constant cost in sequential first-price auctions, winning bids are higher in non-revelation of winning bids than when winning bids are revealed. In contrast, high-cost suppliers have better chances of winning with non-revelation of winning bids because their cost is protected and low-cost suppliers will likely inflate their bids. Although our results suggest that revelation of winning bids is beneficial from a buyer's perspective, the results may change with a larger number of players, or with consideration of quality of the product. Further controlled experiments need to be conducted to analyze the behavioral implications of such settings.

THIS PAGE INTENTIONALLY LEFT BLANK

Summary

The proposed approach, Computing-without-Revealing (CWR), supports research in *information systems* and *risk management*. Our approach also complements, but does not replace, research in economic mechanism design. While mechanism design is focused on truthful revelation through the design of incentives, our approach focuses on protecting confidential information in any mechanism. In this study, we developed new dedicated CWR protocols suited for eProcurement and demonstrated the application of these protocols for the procurement of standard and innovative products.

We proposed the CWR-first price reverse auction, which enables a procurer to “cherry pick” those suppliers who provide the best price for each individual item and thereby lower procurement costs. Such lowering of acquisition costs for procurers will increase their efficiency because they will be able to achieve more with the same financial resources. Suppliers who participate will not see their competitive advantage erode due to the very fact that they participated (e.g., currently, a cost advantage for some components quickly erodes once it becomes known). The eProcurement platforms based on the proposed approach will considerably mitigate the threat of data breach originating from business partners because the approach makes it possible to achieve the desired collaborative goals with business partners without revealing to them the confidential data on which the collaboration depends.

A test bed was developed to compare the performance of CWR-based protocols with the previous-best approaches. Experimental results show that the CWR protocols performed better than previous-best approaches. With this, we conclude that CWR based auctions are lightweight, scalable, and secure.

Finally, we performed a pilot laboratory experiment to understand the behavioral implications of using CWR in an online auction. From the pilot study, we observed that subjects increase their bids based on whether they win in past auctions. Such bid inflation behavior is influenced by hiding the outcomes of the auctions. Further studies in this direction are required to understand the scenarios that enable suppliers towards truthful revelations.

References

Wang, S., Bhandari, S., Chaduvula, S. C., Atallah, M. J., Panchal, J. H., & Ramani, K. (2017). Secure collaboration in engineering systems design. Journal of Computing and Information Science in Engineering, 17(4), 041010.

Wang, S., Nassar, M., Atallah, M., & Malluhi, Q. (2013). Secure and private outsourcing of shape-based feature extraction. In International Conference on Information and Communications Security (pp. 90-99). Springer, Cham.

Kaestner S., Arndt C., & Dillon-Merrill R. (2016). The Cybersecurity Challenge in Acquisition (SYM-AM-16-041). In Proceedings of the thirteenth Annual Research Symposium, Volume 1. https://calhoun.nps.edu/handle/10945/53480.

Ben-David, Assaf, Noam Nisan, and Benny Pinkas. "FairplayMP: a system for secure multi-party computation." In Proceedings of the 15th ACM conference on Computer and communications security, pp. 257-266. ACM, 2008.

Bogetoft, Peter, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen et al. "Secure multiparty computation goes live." In Financial Cryptography and Data Security, pp. 325-343. Springer Berlin Heidelberg, 2009.

Wang, Shumiao, Mohamed Nassar, Mikhail Atallah, and Qutaibah Malluhi. "Secure and private outsourcing of shape-based feature extraction." In Information and Communications Security, pp. 90-99. Springer International Publishing, 2013.

Kaestner Sonia, Arndt Craig, Dillon-Merrill Robin. “The Cybersecurity Challenge in Acquisition”. In Proceedings of the Thirteenth Annual Acquisition Research Symposium, 2016.

Deshpande, Vinayak, Leroy B. Schwarz, Mikhail J. Atallah, Marina Blanton, and Keith B. Frikken. "Outsourcing Manufacturing: Secure Price-Masking Mechanisms for Purchasing Component Parts." Production and Operations Management 20, no. 2 (2011): 165-180.

Coughlan, Peter, William Gates, and Jennifer Lamping. Innovations in defense acquisition auctions: Lessons learned and alternative mechanism designs. No. NPS-AM-08-013. Naval Postgraduate School Monterey CA Graduate School of Business and Public, 2008.

Paillier, Pascal. (1999, May). Public-key cryptosystems based on composite degree residuosity classes. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 223-238). Springer, Berlin, Heidelberg.

Chen, D. L., Schonger, M., & Wickens, C. (2016). oTree—An open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance*, *9*, 88-97.

Ponemon Institute. (2016, Aug). Closing Security Gaps to Protect Corporate Data: A Study of US and European Organizations. https://info.varonis.com/hubfs/docs/research_reports/Varonis_Ponemon_2016_Report.pdf

Acquisition Research Program
Graduate School of Business & Public Policy
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net