02 INFORMATION ABOUT PRINCIPAL INVESTIGATORS/PROJECT DIRECTORS(PI/PD) and co-PRINCIPAL INVESTIGATORS/co-PROJECT DIRECTORS

Submit only ONE copy of this form **for each PI/PD and co-PI/PD** identified on the proposal. The form(s) should be attached to the original proposal as specified in GPG Section II.C.a. Submission of this information is voluntary and is not a precondition of award. This information will not be disclosed to external peer reviewers. ***DO NOT INCLUDE THIS FORM WITH ANY OF THE OTHER COPIES OF YOUR PROPOSAL AS THIS MAY COMPROMISE THE CONFIDENTIALITY OF THE INFORMATION.***

PI/PD Name: Taesoo Kim

Gender: ☒ Male ☐ Female

Ethnicity: (Choose one response) ☐ Hispanic or Latino ☒ Not Hispanic or Latino

Race: (Select one or more)
- ☐ American Indian or Alaska Native
- ☒ Asian
- ☐ Black or African American
- ☐ Native Hawaiian or Other Pacific Islander
- ☐ White

Disability Status: (Select one or more)
- ☐ Hearing Impairment
- ☐ Visual Impairment
- ☐ Mobility/Orthopedic Impairment
- ☐ Other
- ☐ None

Citizenship: (Choose one) ☐ U.S. Citizen ☒ Permanent Resident ☐ Other non-U.S. Citizen

Check here if you do not wish to provide any or all of the above information (excluding PI/PD name): ☒

REQUIRED: Check here if you are currently serving (or have previously served) as a PI, co-PI or PD on any federally funded project ☐

Ethnicity Definition:
Hispanic or Latino. A person of Mexican, Puerto Rican, Cuban, South or Central American, or other Spanish culture or origin, regardless of race.
Race Definitions:
American Indian or Alaska Native. A person having origins in any of the original peoples of North and South America (including Central America), and who maintains tribal affiliation or community attachment.
Asian. A person having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent including, for example, Cambodia, China, India, Japan, Korea, Malaysia, Pakistan, the Philippine Islands, Thailand, and Vietnam.
Black or African American. A person having origins in any of the black racial groups of Africa.
Native Hawaiian or Other Pacific Islander. A person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands.
White. A person having origins in any of the original peoples of Europe, the Middle East, or North Africa.

WHY THIS INFORMATION IS BEING REQUESTED:

The Federal Government has a continuing commitment to monitor the operation of its review and award processes to identify and address any inequities based on gender, race, ethnicity, or disability of its proposed PIs/PDs. To gather information needed for this important task, the proposer should submit a single copy of this form for each identified PI/PD with each proposal. Submission of the requested information is voluntary and will not affect the organization's eligibility for an award. However, information not submitted will seriously undermine the statistical validity, and therefore the usefulness, of information recieved from others. Any individual not wishing to submit some or all the information should check the box provided for this purpose. (The exceptions are the PI/PD name and the information about prior Federal support, the last question above.)

Collection of this information is authorized by the NSF Act of 1950, as amended, 42 U.S.C. 1861, et seq. Demographic data allows NSF to gauge whether our programs and other opportunities in science and technology are fairly reaching and benefiting everyone regardless of demographic category; to ensure that those in under-represented groups have the same knowledge of and access to programs and other research and educational oppurtunities; and to assess involvement of international investigators in work supported by NSF. The information may be disclosed to government contractors, experts, volunteers and researchers to complete assigned work; and to other government agencies in order to coordinate and assess programs. The information may be added to the Reviewer file and used to select potential candidates to serve as peer reviewers or advisory committee members. See Systems of Records, NSF-50, "Principal Investigator/Proposal File and Associated Records", 63 Federal Register 267 (January 5, 1998), and NSF-51, "Reviewer/Proposal File and Associated Records", 63 Federal Register 268 (January 5, 1998).

02 INFORMATION ABOUT PRINCIPAL INVESTIGATORS/PROJECT DIRECTORS(PI/PD) and co-PRINCIPAL INVESTIGATORS/co-PROJECT DIRECTORS

Submit only ONE copy of this form **for each PI/PD and co-PI/PD** identified on the proposal. The form(s) should be attached to the original proposal as specified in GPG Section II.C.a. Submission of this information is voluntary and is not a precondition of award. This information will not be disclosed to external peer reviewers. ***DO NOT INCLUDE THIS FORM WITH ANY OF THE OTHER COPIES OF YOUR PROPOSAL AS THIS MAY COMPROMISE THE CONFIDENTIALITY OF THE INFORMATION.***

PI/PD Name: Zhiqiang Lin

Gender: ☒ Male ☐ Female

Ethnicity: (Choose one response) ☐ Hispanic or Latino ☒ Not Hispanic or Latino

Race: (Select one or more)
- ☐ American Indian or Alaska Native
- ☒ Asian
- ☐ Black or African American
- ☐ Native Hawaiian or Other Pacific Islander
- ☐ White

Disability Status: (Select one or more)
- ☐ Hearing Impairment
- ☐ Visual Impairment
- ☐ Mobility/Orthopedic Impairment
- ☐ Other
- ☒ None

Citizenship: (Choose one) ☐ U.S. Citizen ☒ Permanent Resident ☐ Other non-U.S. Citizen

Check here if you do not wish to provide any or all of the above information (excluding PI/PD name): ☐

REQUIRED: Check here if you are currently serving (or have previously served) as a PI, co-PI or PD on any federally funded project ☒

Ethnicity Definition:
Hispanic or Latino. A person of Mexican, Puerto Rican, Cuban, South or Central American, or other Spanish culture or origin, regardless of race.

Race Definitions:
American Indian or Alaska Native. A person having origins in any of the original peoples of North and South America (including Central America), and who maintains tribal affiliation or community attachment.
Asian. A person having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent including, for example, Cambodia, China, India, Japan, Korea, Malaysia, Pakistan, the Philippine Islands, Thailand, and Vietnam.
Black or African American. A person having origins in any of the black racial groups of Africa.
Native Hawaiian or Other Pacific Islander. A person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands.
White. A person having origins in any of the original peoples of Europe, the Middle East, or North Africa.

WHY THIS INFORMATION IS BEING REQUESTED:

The Federal Government has a continuing commitment to monitor the operation of its review and award processes to identify and address any inequities based on gender, race, ethnicity, or disability of its proposed PIs/PDs. To gather information needed for this important task, the proposer should submit a single copy of this form for each identified PI/PD with each proposal. Submission of the requested information is voluntary and will not affect the organization's eligibility for an award. However, information not submitted will seriously undermine the statistical validity, and therefore the usefulness, of information recieved from others. Any individual not wishing to submit some or all the information should check the box provided for this purpose. (The exceptions are the PI/PD name and the information about prior Federal support, the last question above.)

Collection of this information is authorized by the NSF Act of 1950, as amended, 42 U.S.C. 1861, et seq. Demographic data allows NSF to gauge whether our programs and other opportunities in science and technology are fairly reaching and benefiting everyone regardless of demographic category; to ensure that those in under-represented groups have the same knowledge of and access to programs and other research and educational oppurtunities; and to assess involvement of international investigators in work supported by NSF. The information may be disclosed to government contractors, experts, volunteers and researchers to complete assigned work; and to other government agencies in order to coordinate and assess programs. The information may be added to the Reviewer file and used to select potential candidates to serve as peer reviewers or advisory committee members. See Systems of Records, NSF-50, "Principal Investigator/Proposal File and Associated Records", 63 Federal Register 267 (January 5, 1998), and NSF-51, "Reviewer/Proposal File and Associated Records", 63 Federal Register 268 (January 5, 1998).

List of Suggested Reviewers or Reviewers Not To Include (optional)

SUGGESTED REVIEWERS:
Not Listed

REVIEWERS NOT TO INCLUDE:
Not Listed

List of Suggested Reviewers or Reviewers Not To Include (optional)

SUGGESTED REVIEWERS:
Not Listed

REVIEWERS NOT TO INCLUDE:
Not Listed

COVER SHEET FOR PROPOSAL TO THE NATIONAL SCIENCE FOUNDATION

PROGRAM ANNOUNCEMENT/SOLICITATION NO./CLOSING DATE/if not in response to a program announcement/solicitation enter NSF 15-1

NSF 15-575 **09/16/15**

FOR CONSIDERATION BY NSF ORGANIZATION UNIT(S) (Indicate the most specific unit known, i.e. program, division, etc.)

CNS - Secure &Trustworthy Cyberspace

FOR NSF USE ONLY

NSF PROPOSAL NUMBER

1563848

DATE RECEIVED	NUMBER OF COPIES	DIVISION ASSIGNED	FUND CODE	DUNS# (Data Universal Numbering System)	FILE LOCATION
09/16/2015	**2**	**05050000 CNS**	**8060**	**097394084**	**10/01/2015 7:34pm S**

EMPLOYER IDENTIFICATION NUMBER (EIN) OR TAXPAYER IDENTIFICATION NUMBER (TIN)

580603146

SHOW PREVIOUS AWARD NO. IF THIS IS
☐ A RENEWAL
☐ AN ACCOMPLISHMENT-BASED RENEWAL

IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY? YES ☐ NO ☒ IF YES, LIST ACRONYM(S)

NAME OF ORGANIZATION TO WHICH AWARD SHOULD BE MADE

Georgia Tech Research Corporation

AWARDEE ORGANIZATION CODE (IF KNOWN)

0015693000

ADDRESS OF AWARDEE ORGANIZATION, INCLUDING 9 DIGIT ZIP CODE

Office of Sponsored Programs
Atlanta, GA 30332-0420

NAME OF PRIMARY PLACE OF PERF

Georgia Institute of Technology

ADDRESS OF PRIMARY PLACE OF PERF, INCLUDING 9 DIGIT ZIP CODE

Georgia Institute of Technology
225 North Avenue
GA ,303320002 ,US.

IS AWARDEE ORGANIZATION (Check All That Apply) (See GPG II.C For Definitions)
☐ SMALL BUSINESS ☐ MINORITY BUSINESS
☐ FOR-PROFIT ORGANIZATION ☐ WOMAN-OWNED BUSINESS

☐ IF THIS IS A PRELIMINARY PROPOSAL THEN CHECK HERE

TITLE OF PROPOSED PROJECT **TWC: Medium: Collaborative: Systems, Tools, and Techniques for Executing, Managing, and Securing SGX Programs**

REQUESTED AMOUNT	PROPOSED DURATION (1-60 MONTHS)	REQUESTED STARTING DATE	SHOW RELATED PRELIMINARY PROPOSAL NO. IF APPLICABLE
$ **671,776**	**48** months	**06/01/16**	

THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW

☐ BEGINNING INVESTIGATOR (GPG I.G.2)
☐ DISCLOSURE OF LOBBYING ACTIVITIES (GPG II.C.1.e)
☐ PROPRIETARY & PRIVILEGED INFORMATION (GPG I.D, II.C.1.d)
☐ HISTORIC PLACES (GPG II.C.2.j)
☐ VERTEBRATE ANIMALS (GPG II.D.6) IACUC App. Date _____
PHS Animal Welfare Assurance Number _____
☒ FUNDING MECHANISM **Research - other than RAPID or EAGER**

☐ HUMAN SUBJECTS (GPG II.D.7) Human Subjects Assurance Number _____
Exemption Subsection _____ or IRB App. Date _____
☐ INTERNATIONAL ACTIVITIES: COUNTRY/COUNTRIES INVOLVED (GPG II.C.2.j)

☒ COLLABORATIVE STATUS
A collaborative proposal from multiple organizations (GPG II.D.4.b)

PI/PD DEPARTMENT
School of Computer Science

PI/PD FAX NUMBER

PI/PD POSTAL ADDRESS
266 Ferst Drive

Atlanta, GA 303320420
United States

NAMES (TYPED)	High Degree	Yr of Degree	Telephone Number	Email Address
PI/PD NAME **Taesoo Kim**	**PhD**	**2014**	**404-894-4819**	**taesoo@gatech.edu**
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				

CERTIFICATION PAGE

Certification for Authorized Organizational Representative (or Equivalent) or Individual Applicant

By electronically signing and submitting this proposal, the Authorized Organizational Representative (AOR) or Individual Applicant is: (1) certifying that statements made herein are true and complete to the best of his/her knowledge; and (2) agreeing to accept the obligation to comply with NSF award terms and conditions if an award is made as a result of this application. Further, the applicant is hereby providing certifications regarding conflict of interest (when applicable), drug-free workplace, debarment and suspension, lobbying activities (see below), nondiscrimination, flood hazard insurance (when applicable), responsible conduct of research, organizational support, Federal tax obligations, unpaid Federal tax liability, and criminal convictions as set forth in the NSF Proposal & Award Policies & Procedures Guide,Part I: the Grant Proposal Guide (GPG). Willful provision of false information in this application and its supporting documents or in reports required under an ensuing award is a criminal offense (U.S. Code, Title 18, Section 1001).

Certification Regarding Conflict of Interest

The AOR is required to complete certifications stating that the organization has implemented and is enforcing a written policy on conflicts of interest (COI), consistent with the provisions of AAG Chapter IV.A.; that, to the best of his/her knowledge, all financial disclosures required by the conflict of interest policy were made; and that conflicts of interest, if any, were, or prior to the organization's expenditure of any funds under the award, will be, satisfactorily managed, reduced or eliminated in accordance with the organization's conflict of interest policy. Conflicts that cannot be satisfactorily managed, reduced or eliminated and research that proceeds without the imposition of conditions or restrictions when a conflict of interest exists, must be disclosed to NSF via use of the Notifications and Requests Module in FastLane.

Drug Free Work Place Certification

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent), is providing the Drug Free Work Place Certification contained in Exhibit II-3 of the Grant Proposal Guide.

Debarment and Suspension Certification

(If answer "yes", please provide explanation.)

Is the organization or its principals presently debarred, suspended, proposed for debarment, declared ineligible, or voluntarily excluded from covered transactions by any Federal department or agency? Yes ☐ No ☒

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) or Individual Applicant is providing the Debarment and Suspension Certification contained in Exhibit II-4 of the Grant Proposal Guide.

Certification Regarding Lobbying

This certification is required for an award of a Federal contract, grant, or cooperative agreement exceeding $100,000 and for an award of a Federal loan or a commitment providing for the United States to insure or guarantee a loan exceeding $150,000.

Certification for Contracts, Grants, Loans and Cooperative Agreements

The undersigned certifies, to the best of his or her knowledge and belief, that:
(1) No Federal appropriated funds have been paid or will be paid, by or on behalf of the undersigned, to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with the awarding of any Federal contract, the making of any Federal grant, the making of any Federal loan, the entering into of any cooperative agreement, and the extension, continuation, renewal, amendment, or modification of any Federal contract, grant, loan, or cooperative agreement.
(2) If any funds other than Federal appropriated funds have been paid or will be paid to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with this Federal contract, grant, loan, or cooperative agreement, the undersigned shall complete and submit Standard Form-LLL, ''Disclosure of Lobbying Activities,'' in accordance with its instructions.
(3) The undersigned shall require that the language of this certification be included in the award documents for all subawards at all tiers including subcontracts, subgrants, and contracts under grants, loans, and cooperative agreements and that all subrecipients shall certify and disclose accordingly.

This certification is a material representation of fact upon which reliance was placed when this transaction was made or entered into. Submission of this certification is a prerequisite for making or entering into this transaction imposed by section 1352, Title 31, U.S. Code. Any person who fails to file the required certification shall be subject to a civil penalty of not less than $10,000 and not more than $100,000 for each such failure.

Certification Regarding Nondiscrimination

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) is providing the Certification Regarding Nondiscrimination contained in Exhibit II-6 of the Grant Proposal Guide.

Certification Regarding Flood Hazard Insurance

Two sections of the National Flood Insurance Act of 1968 (42 USC §4012a and §4106) bar Federal agencies from giving financial assistance for acquisition or construction purposes in any area identified by the Federal Emergency Management Agency (FEMA) as having special flood hazards unless the:
(1) community in which that area is located participates in the national flood insurance program; and
(2) building (and any related equipment) is covered by adequate flood insurance.

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) or Individual Applicant located in FEMA-designated special flood hazard areas is certifying that adequate flood insurance has been or will be obtained in the following situations:
(1) for NSF grants for the construction of a building or facility, regardless of the dollar amount of the grant; and
(2) for other NSF grants when more than $25,000 has been budgeted in the proposal for repair, alteration or improvement (construction) of a building or facility.

Certification Regarding Responsible Conduct of Research (RCR)
(This certification is not applicable to proposals for conferences, symposia, and workshops.)

By electronically signing the Certification Pages, the Authorized Organizational Representative is certifying that, in accordance with the NSF Proposal & Award Policies & Procedures Guide, Part II, Award & Administration Guide (AAG) Chapter IV.B., the institution has a plan in place to provide appropriate training and oversight in the responsible and ethical conduct of research to undergraduates, graduate students and postdoctoral researchers who will be supported by NSF to conduct research. The AOR shall require that the language of this certification be included in any award documents for all subawards at all tiers.

CERTIFICATION PAGE - CONTINUED

Certification Regarding Organizational Support

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) is certifying that there is organizational support for the proposal as required by Section 526 of the America COMPETES Reauthorization Act of 2010. This support extends to the portion of the proposal developed to satisfy the Broader Impacts Review Criterion as well as the Intellectual Merit Review Criterion, and any additional review criteria specified in the solicitation. Organizational support will be made available, as described in the proposal, in order to address the broader impacts and intellectual merit activities to be undertaken.

Certification Regarding Federal Tax Obligations

When the proposal exceeds $5,000,000, the Authorized Organizational Representative (or equivalent) is required to complete the following certification regarding Federal tax obligations. By electronically signing the Certification pages, the Authorized Organizational Representative is certifying that, to the best of their knowledge and belief, the proposing organization:
(1) has filed all Federal tax returns required during the three years preceding this certification;
(2) has not been convicted of a criminal offense under the Internal Revenue Code of 1986; and
(3) has not, more than 90 days prior to this certification, been notified of any unpaid Federal tax assessment for which the liability remains unsatisfied, unless the assessment is the subject of an installment agreement or offer in compromise that has been approved by the Internal Revenue Service and is not in default, or the assessment is the subject of a non-frivolous administrative or judicial proceeding.

Certification Regarding Unpaid Federal Tax Liability

When the proposing organization is a corporation, the Authorized Organizational Representative (or equivalent) is required to complete the following certification regarding Federal Tax Liability:

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) is certifying that the corporation has no unpaid Federal tax liability that has been assessed, for which all judicial and administrative remedies have been exhausted or lapsed, and that is not being paid in a timely manner pursuant to an agreement with the authority responsible for collecting the tax liability.

Certification Regarding Criminal Convictions

When the proposing organization is a corporation, the Authorized Organizational Representative (or equivalent) is required to complete the following certification regarding Criminal Convictions:

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) is certifying that the corporation has not been convicted of a felony criminal violation under any Federal law within the 24 months preceding the date on which the certification is signed.

AUTHORIZED ORGANIZATIONAL REPRESENTATIVE	SIGNATURE	DATE
NAME **Tanya Blackwell**	**Electronic Signature**	**Sep 16 2015 1:05PM**

TELEPHONE NUMBER	EMAIL ADDRESS	FAX NUMBER
	Tanya.Blackwell@osp.gatech.edu	

COVER SHEET FOR PROPOSAL TO THE NATIONAL SCIENCE FOUNDATION

PROGRAM ANNOUNCEMENT/SOLICITATION NO./CLOSING DATE/if not in response to a program announcement/solicitation enter NSF 15-1

NSF 15-575 **09/16/15**

FOR CONSIDERATION BY NSF ORGANIZATION UNIT(S) (Indicate the most specific unit known, i.e. program, division, etc.)

CNS - Secure &Trustworthy Cyberspace

FOR NSF USE ONLY

NSF PROPOSAL NUMBER

1564112

DATE RECEIVED	NUMBER OF COPIES	DIVISION ASSIGNED	FUND CODE	DUNS# (Data Universal Numbering System)	FILE LOCATION
09/16/2015	**2**	**05050000 CNS**	**8060**	**800188161**	**10/01/2015 7:34pm S**

EMPLOYER IDENTIFICATION NUMBER (EIN) OR TAXPAYER IDENTIFICATION NUMBER (TIN)

751305566

SHOW PREVIOUS AWARD NO. IF THIS IS
☐ A RENEWAL
☐ AN ACCOMPLISHMENT-BASED RENEWAL

IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY? YES ☐ NO ☒ IF YES, LIST ACRONYM(S)

NAME OF ORGANIZATION TO WHICH AWARD SHOULD BE MADE

University of Texas at Dallas

AWARDEE ORGANIZATION CODE (IF KNOWN)

0035691000

ADDRESS OF AWARDEE ORGANIZATION, INCLUDING 9 DIGIT ZIP CODE

University of Texas at Dallas
800 W. Campbell Rd., AD15
Richardson, TX. 750803021

NAME OF PRIMARY PLACE OF PERF

University of Texas at Dallas

ADDRESS OF PRIMARY PLACE OF PERF, INCLUDING 9 DIGIT ZIP CODE

University of Texas at Dallas
800 W Campbell
Richardson ,TX ,750803021 ,US.

IS AWARDEE ORGANIZATION (Check All That Apply) (See GPG II.C For Definitions)
☐ SMALL BUSINESS ☐ MINORITY BUSINESS ☐ IF THIS IS A PRELIMINARY PROPOSAL
☐ FOR-PROFIT ORGANIZATION ☐ WOMAN-OWNED BUSINESS THEN CHECK HERE

TITLE OF PROPOSED PROJECT **TWC: Medium: Collaborative: Systems, Tools, and Techniques for Executing, Managing, and Securing SGX Programs**

REQUESTED AMOUNT	PROPOSED DURATION (1-60 MONTHS)	REQUESTED STARTING DATE	SHOW RELATED PRELIMINARY PROPOSAL NO. IF APPLICABLE
$ **528,223**	**48** months	**06/01/16**	

THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW

☐ BEGINNING INVESTIGATOR (GPG I.G.2)
☐ DISCLOSURE OF LOBBYING ACTIVITIES (GPG II.C.1.e)
☐ PROPRIETARY & PRIVILEGED INFORMATION (GPG I.D, II.C.1.d)
☐ HISTORIC PLACES (GPG II.C.2.j)
☐ VERTEBRATE ANIMALS (GPG II.D.6) IACUC App. Date _____
PHS Animal Welfare Assurance Number _____
☒ FUNDING MECHANISM **Research - other than RAPID or EAGER**

☐ HUMAN SUBJECTS (GPG II.D.7) Human Subjects Assurance Number _____
Exemption Subsection _____ or IRB App. Date _____
☐ INTERNATIONAL ACTIVITIES: COUNTRY/COUNTRIES INVOLVED (GPG II.C.2.j)

☒ COLLABORATIVE STATUS
A collaborative proposal from multiple organizations (GPG II.D.4.b)

PI/PD DEPARTMENT
Computer Sciences

PI/PD POSTAL ADDRESS
800 W. Campbell Rd.
Richardson, TX 750803021
United States

PI/PD FAX NUMBER
972-883-2399

NAMES (TYPED)	High Degree	Yr of Degree	Telephone Number	Email Address
PI/PD NAME **Zhiqiang Lin**	**DSc**	**2011**	**972-883-4244**	**zhiqiang.lin@utdallas.edu**
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				

CERTIFICATION PAGE

Certification for Authorized Organizational Representative (or Equivalent) or Individual Applicant

By electronically signing and submitting this proposal, the Authorized Organizational Representative (AOR) or Individual Applicant is: (1) certifying that statements made herein are true and complete to the best of his/her knowledge; and (2) agreeing to accept the obligation to comply with NSF award terms and conditions if an award is made as a result of this application. Further, the applicant is hereby providing certifications regarding conflict of interest (when applicable), drug-free workplace, debarment and suspension, lobbying activities (see below), nondiscrimination, flood hazard insurance (when applicable), responsible conduct of research, organizational support, Federal tax obligations, unpaid Federal tax liability, and criminal convictions as set forth in the NSF Proposal & Award Policies & Procedures Guide,Part I: the Grant Proposal Guide (GPG). Willful provision of false information in this application and its supporting documents or in reports required under an ensuing award is a criminal offense (U.S. Code, Title 18, Section 1001).

Certification Regarding Conflict of Interest

The AOR is required to complete certifications stating that the organization has implemented and is enforcing a written policy on conflicts of interest (COI), consistent with the provisions of AAG Chapter IV.A.; that, to the best of his/her knowledge, all financial disclosures required by the conflict of interest policy were made; and that conflicts of interest, if any, were, or prior to the organization's expenditure of any funds under the award, will be, satisfactorily managed, reduced or eliminated in accordance with the organization's conflict of interest policy. Conflicts that cannot be satisfactorily managed, reduced or eliminated and research that proceeds without the imposition of conditions or restrictions when a conflict of interest exists, must be disclosed to NSF via use of the Notifications and Requests Module in FastLane.

Drug Free Work Place Certification

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent), is providing the Drug Free Work Place Certification contained in Exhibit II-3 of the Grant Proposal Guide.

Debarment and Suspension Certification

(If answer "yes", please provide explanation.)

Is the organization or its principals presently debarred, suspended, proposed for debarment, declared ineligible, or voluntarily excluded from covered transactions by any Federal department or agency? Yes ☐ No ☒

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) or Individual Applicant is providing the Debarment and Suspension Certification contained in Exhibit II-4 of the Grant Proposal Guide.

Certification Regarding Lobbying

This certification is required for an award of a Federal contract, grant, or cooperative agreement exceeding $100,000 and for an award of a Federal loan or a commitment providing for the United States to insure or guarantee a loan exceeding $150,000.

Certification for Contracts, Grants, Loans and Cooperative Agreements

The undersigned certifies, to the best of his or her knowledge and belief, that:
(1) No Federal appropriated funds have been paid or will be paid, by or on behalf of the undersigned, to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with the awarding of any Federal contract, the making of any Federal grant, the making of any Federal loan, the entering into of any cooperative agreement, and the extension, continuation, renewal, amendment, or modification of any Federal contract, grant, loan, or cooperative agreement.
(2) If any funds other than Federal appropriated funds have been paid or will be paid to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with this Federal contract, grant, loan, or cooperative agreement, the undersigned shall complete and submit Standard Form-LLL, ''Disclosure of Lobbying Activities,'' in accordance with its instructions.
(3) The undersigned shall require that the language of this certification be included in the award documents for all subawards at all tiers including subcontracts, subgrants, and contracts under grants, loans, and cooperative agreements and that all subrecipients shall certify and disclose accordingly.

This certification is a material representation of fact upon which reliance was placed when this transaction was made or entered into. Submission of this certification is a prerequisite for making or entering into this transaction imposed by section 1352, Title 31, U.S. Code. Any person who fails to file the required certification shall be subject to a civil penalty of not less than $10,000 and not more than $100,000 for each such failure.

Certification Regarding Nondiscrimination

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) is providing the Certification Regarding Nondiscrimination contained in Exhibit II-6 of the Grant Proposal Guide.

Certification Regarding Flood Hazard Insurance

Two sections of the National Flood Insurance Act of 1968 (42 USC §4012a and §4106) bar Federal agencies from giving financial assistance for acquisition or construction purposes in any area identified by the Federal Emergency Management Agency (FEMA) as having special flood hazards unless the:
(1) community in which that area is located participates in the national flood insurance program; and
(2) building (and any related equipment) is covered by adequate flood insurance.

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) or Individual Applicant located in FEMA-designated special flood hazard areas is certifying that adequate flood insurance has been or will be obtained in the following situations:
(1) for NSF grants for the construction of a building or facility, regardless of the dollar amount of the grant; and
(2) for other NSF grants when more than $25,000 has been budgeted in the proposal for repair, alteration or improvement (construction) of a building or facility.

Certification Regarding Responsible Conduct of Research (RCR)
(This certification is not applicable to proposals for conferences, symposia, and workshops.)

By electronically signing the Certification Pages, the Authorized Organizational Representative is certifying that, in accordance with the NSF Proposal & Award Policies & Procedures Guide, Part II, Award & Administration Guide (AAG) Chapter IV.B., the institution has a plan in place to provide appropriate training and oversight in the responsible and ethical conduct of research to undergraduates, graduate students and postdoctoral researchers who will be supported by NSF to conduct research. The AOR shall require that the language of this certification be included in any award documents for all subawards at all tiers.

CERTIFICATION PAGE - CONTINUED

Certification Regarding Organizational Support

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) is certifying that there is organizational support for the proposal as required by Section 526 of the America COMPETES Reauthorization Act of 2010. This support extends to the portion of the proposal developed to satisfy the Broader Impacts Review Criterion as well as the Intellectual Merit Review Criterion, and any additional review criteria specified in the solicitation. Organizational support will be made available, as described in the proposal, in order to address the broader impacts and intellectual merit activities to be undertaken.

Certification Regarding Federal Tax Obligations

When the proposal exceeds $5,000,000, the Authorized Organizational Representative (or equivalent) is required to complete the following certification regarding Federal tax obligations. By electronically signing the Certification pages, the Authorized Organizational Representative is certifying that, to the best of their knowledge and belief, the proposing organization:
(1) has filed all Federal tax returns required during the three years preceding this certification;
(2) has not been convicted of a criminal offense under the Internal Revenue Code of 1986; and
(3) has not, more than 90 days prior to this certification, been notified of any unpaid Federal tax assessment for which the liability remains unsatisfied, unless the assessment is the subject of an installment agreement or offer in compromise that has been approved by the Internal Revenue Service and is not in default, or the assessment is the subject of a non-frivolous administrative or judicial proceeding.

Certification Regarding Unpaid Federal Tax Liability

When the proposing organization is a corporation, the Authorized Organizational Representative (or equivalent) is required to complete the following certification regarding Federal Tax Liability:

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) is certifying that the corporation has no unpaid Federal tax liability that has been assessed, for which all judicial and administrative remedies have been exhausted or lapsed, and that is not being paid in a timely manner pursuant to an agreement with the authority responsible for collecting the tax liability.

Certification Regarding Criminal Convictions

When the proposing organization is a corporation, the Authorized Organizational Representative (or equivalent) is required to complete the following certification regarding Criminal Convictions:

By electronically signing the Certification Pages, the Authorized Organizational Representative (or equivalent) is certifying that the corporation has not been convicted of a felony criminal violation under any Federal law within the 24 months preceding the date on which the certification is signed.

AUTHORIZED ORGANIZATIONAL REPRESENTATIVE	SIGNATURE	DATE
NAME **Emily Lacy**	**Electronic Signature**	**Sep 16 2015 5:33PM**

TELEPHONE NUMBER	EMAIL ADDRESS	FAX NUMBER
	emily.lacy@utdallas.edu	

TWC: Medium: Collaborative: Systems, Tools, and Techniques for Executing, Managing, and Securing SGX Programs

The Intel Software Guard Extensions (SGX)—a game-changing feature introduced in the recent Intel Skylake CPU—is a new technology likely to make secure and trustworthy computing in a hostile environment practical. At a high level, SGX consists of a set of new instructions that can be used to create secure regions (i.e., enclaves) to defeat attacks that aim to steal or tamper with the data within an enclave. Without a doubt, we expect that SGX will allow developers to protect sensitive code and data from unauthorized access or modification by software running at higher privilege levels such as an OS or a hypervisor.

However, SGX is merely a set of instructions; it lacks support from the OS and libraries. These deficiencies allow programmers to easily introduce naive yet preventable bugs that often lead to critical security holes in an enclave program. Further, designing a correct and secure SGX infrastructure is also far from straightforward: enclave programs rely on the support of an underlying OS, but their security models exclude the OS from the TCB. This unconventional dependency makes various attack vectors, which are often considered impractical in a traditional setting, immediate and practical, especially in a cloud environment. Meanwhile, using SGX does not secure vulnerable enclave programs automatically; typical threats and attack vectors such as buffer overflows still exist. In fact, the compromise of an enclave program becomes more critical than before, because SGX makes it impossible to analyze or even detect such compromises, and also, for end users, a compromised execution is completely indistinguishable from a correct execution. Therefore, we believe now is the proper time to thoroughly revisit what we have learned from trusted computing within the context of Intel SGX. In this proposed project, we would like to investigate the infrastructure, tools, and libraries necessary to securely support SGX.

Intellectual Merit. One hurdle to the applicability of SGX is the lack of toolchain or system software support. Much like Xen/KVM for Intel VT-X, one important advancement of this proposed research is that we will investigate how to enable application developers to securely use the SGX instruction set, with toolchain, programming abstractions (e.g., library), and operating system support (e.g., system calls). Most importantly, we will perform our research in an open, community-driven fashion to foster the adoption of Intel SGX. All of our efforts will be integrated into the GNU/Linux platform.

Another important advancement of this proposed research is that we will investigate techniques to defeat various attacks against SGX programs. Specifically, since SGX programs will be running in a completely hostile environment, they will encounter many attacks, such as system call level attacks or side channel attacks. Furthermore, SGX programs may not be vulnerability free and can still be exploitable. Finally, SGX can also be used by malicious software to build irreversible malware. In this proposed research, we will systematically study these attacks and explore the systems and software defenses necessary to secure the SGX programs from the enclave itself and defeat the malicious use of SGX from the underlying OS.

Broader Impacts. This research will develop a toolchain and system support to enable GNU/Linux application programmers to use SGX securely, while also providing a holistic solution to secure SGX programs against various attacks and defending SGX against malicious use. These results are not only applicable to the broader scope of software development in general, but also are of special interest to security. In addition to the broader academic impact, the proposed work is also of particular interest to the open source community as well as the systems and security industry. The PIs have included a collaboration letter from Intel, indicating their strong interest in this project.

The proposed research results will be integrated into security curricula with lecture notes and hands-on labs. These materials will be made freely available to the public. The PIs are also engaged in outreach activities, including partnering with the local (HBCU) colleges, organizing cybersecurity workshops in K-12 summer camps and an SGX workshop in academic conferences, and collaborating with systems and security companies.

TABLE OF CONTENTS

For font size and page formatting specifications, see GPG section II.B.2.

*Proposers may select any numbering mechanism for the proposal. The entire proposal however, must be paginated. Complete both columns only if the proposal is numbered consecutively.

TABLE OF CONTENTS

For font size and page formatting specifications, see GPG section II.B.2.

*Proposers may select any numbering mechanism for the proposal. The entire proposal however, must be paginated. Complete both columns only if the proposal is numbered consecutively.

D.1 Introduction

Background. Trusted computing, or the Trusted Execution Environment (TEE), is the foundational technology that ensures confidentiality and integrity in modern computing. Over the past few decades, a considerable amount of research has been carried out to search for practical ways to achieve trusted computing, e.g., using a formally verified operating system (OS) [47], virtual machine monitor (VMM) or hypervisor [22, 53, 56, 74, 76, 91], system management mode (SMM) [82], and BIOS [77], or hardware assistance [55]. Increasingly, hardware technologies for TEE (e.g., TPM [64], ARM Trusted Zone [70], Intel TXT, and AMD SVM [80])—already widely deployed in consumer and enterprise products— have rapidly matured, especially with the commoditization of Intel Software Guard Extensions (SGX) [10, 39, 57], which became market available in August 2015 in Intel's Skylake CPU.

At a high level, Intel SGX allows an application or part of an application to run inside a secure *enclave*, which is an isolated execution environment. Intel SGX hardware, as a part of the CPU, protects the enclave against malicious software, including the operating system, hypervisor, or even low-level firmware code (e.g., SMM) from compromising its integrity. This *isolation* enabled by Intel SGX is particularly useful in cloud computing environments, where customers cannot control the infrastructure owned by cloud providers. Consequently, initial exploration into secure execution has begun for cloud settings. Haven [11] pioneered the idea of enabling unmodified application binaries to run on Intel SGX by utilizing an OS library [65]. VC3 [72] suggested privacy-aware data analytics in the cloud. As witnessed by these early efforts, we believe that Intel SGX can be used to achieve unprecedented security for many cloud applications (and beyond), thereby safely taking advantage of the flexibility, elasticity, and economy-of-scale provided by cloud infrastructures.

At a low level, Intel SGX is fundamentally an extension to the x86 instruction set architecture (ISA) that enables an application to instantiate one or more isolated enclaves. SGX is implemented largely as two components within a regular CPU: a memory controller that regulates external memory accesses and also performs encryption/decryption of their contents; and an attestation engine that is mainly in charge of remote attestation and sealing of enclaves, which enables verification of enclave integrity. Accordingly, Intel SGX consists of two large sets of new instructions; those utilizing memory management for *isolation*, and those providing security for *attestation*, marked respectively as `MEM` and `SEC` in Table 1).

P	Type	Instruction	Description	V
P	`MEM`	`EADD`	Add a page	r1
P	`MEM`	`EBLOCK`	Block an EPC page	r1
P	`EXE`	`ECREATE`	Create an enclave	r1
P	`DBG`	`EDBGRD`	Read data by debugger	r1
P	`DBG`	`EDBGWR`	Write data by debugger	r1
P	`MEM`	`EEXTEND`	Extend EPC page measurement	r1
P	`EXE`	`EINIT`	Initialize an enclave	r1
P	`MEM`	`ELDB`	Load an EPC page as blocked	r1
P	`MEM`	`ELDU`	Load an EPC page as unblocked	r1
P	`SEC`	`EPA`	Add version array	r1
P	`MEM`	`EREMOVE`	Remove a page from EPC	r1
P	`MEM`	`ETRACK`	Activate EBLOCK checks	r1
P	`MEM`	`EWB`	Write back/invalidate an EPC page	r1
P	`MEM`	`EAUG`	Allocate page to an existing enclave	r2
P	`SEC`	`EMODPR`	Restrict page permissions	r2
P	`EXE`	`EMODT`	Change the type of an EPC page	r2
U	`EXE`	`EENTER`	Enter an enclave	r1
U	`EXE`	`EEXIT`	Exit an enclave	r1
U	`SEC`	`EGETKEY`	Create a cryptographic key	r1
U	`SEC`	`EREPORT`	Create a cryptographic report	r1
U	`EXE`	`ERESUME`	Re-enter an enclave	r1
U	`MEM`	`EACCEPT`	Accept changes to a page	r2
U	`SEC`	`EMODPE`	Enhance access rights	r2
U	`MEM`	`EACCEPTCOPY`	Copy page to a new location	r2

Table 1: SGX Instruction Overview. P: Privileged (ring 0) instructions; U: User-level (ring 3) instructions; V: Version; r1: Revision 1 [42]; r2: Revision 2 [43]; MEM: Memory management related; EXE: Enclave execution related; SEC: Security or permissions related.

When the processor accesses enclave data, it automatically transfers to a new CPU mode, called *enclave mode*. The enclave mode enforces additional hardware checks on each memory access, such that only code inside the enclave can access its own enclave region. That is, memory access from both non-enclaves and different enclaves is prohibited. Note that the memory access policy on non-enclave regions remains the same, i.e., a traditional page walk is performed for both accesses from non-enclaves and enclaves to non-enclave memory. The enclave data is stored in a reserved memory region called the Enclave Page Cache (EPC). To defend against known memory attacks such as memory snooping, memory content in the EPC is encrypted by the Memory Encryption Engine (MEE). The memory content in

the EPC is decrypted only when entering the CPU package, where the code and data are protected by the *enclave mode*, and then re-encrypted when leaving the CPU back to the EPC memory region.

For programmers to use these hardware features, SGX introduces a set of instructions (Table 1) and data structures (Table 2) to support enclave and EPC-related operations. Instructions are classified into user-level instructions (ring 3) and privileged instructions (ring 0). The families of user-level and privileged instructions are called `ENCLU` and `ENCLS`, respectively. For example, the user-level instruction `EENTER` allows the host program to transfer control to an existing enclave program, while `ECREATE` is a privileged instruction that allocates available EPC pages for a new enclave.

Instruction		Description
`EPCM`	Enclave Page Cache Map	Meta-data of an EPC page
`SECS`	Enclave Control Structure	Meta-data of an enclave
`TCS`	Thread Control Structure	Meta-data of a single thread
`SSA`	State Save Area	Used to save processor state
`PageInfo`	Page Information	Used for EPC-management
`SECINFO`	Security Information	Meta-data of an enclave page
`PCMD`	Paging Crypto MetaData	Used to track a page-out page
`SIGSTRUCT`	Enclave Signature Structure	Enclave certificate
`EINITTOKEN`	EINIT Token Structure	Used to validate the enclave
`REPORT`	Report Structure	Return structure of `EREPORT`
`TARGETINFO`	Report Target Info	Parameter for `EREPORT`
`KEYREQUEST`	Key Request	Parameter for `EGETKEY`
`VA`	Version Array	Version for evicted EPC pages

Table 2: Hardware Level Data Structure in Intel SGX.

Motivation. The security of SGX programs does not come for free, but rather depends upon correct usage of SGX instructions. Unfortunately, it is non-trivial for programmers to correctly use such instructions. Worse yet, there is no way to guarantee or validate the correctness of SGX programs. Application programmers should therefore avoid using individual instructions, and instead utilize higher-level abstractions that thoroughly handle the security pitfalls or subtleties of the SGX programming. Although SGX is being integrated into commodity hardware, APIs and system infrastructure that are designed to utilize SGX need to be developed. This includes operating system support, packaging and distribution, debugging and monitoring, and even toolchains for the development of SGX programs. For example, many SGX instructions require kernel-level privileges (ring 0), but system call interface and operating system service/support for SGX has not yet been implemented.

While it may appear that Intel should have led such efforts to develop systems and software support, we note that there is a large demand for a community-driven alternative. If we consider the development and adoption of Intel VT-x, we notice that there are many successful open platforms built around these technologies, such as the Xen and KVM projects. At the same time, closed-source solutions, such as Microsoft HyperV and VMware vSphere, also exist. Because commercial hardware or software vendors might not always be willing to release source code utilizing SGX, and recognizing the benefit of open-source systems utilizing hardware-based ISA extensions, we believe there is a need to develop SGX support in an open manner. We believe an open-source ecosystem surrounding SGX will foster continued research by the security community, and ensure that these extensions are used to further secure x86-based computing environments.

SGX enclave programs will certainly be attacked from various vectors, such as a malicious OS, side channel attacks, or remotely exploitable vulnerabilities. In particular, since SGX programs will be running in a completely hostile environment, it is unclear how to defeat malicious operating system attacks such as system call level attacks (e.g., Iago [21] attack), the Cuckoo attack [63], or controlled side channel attacks [87]. Furthermore, SGX-based bugs are critical in terms of security, and the compromise of an enclave program becomes more critical than before. Also, SGX makes it impossible to analyze or even detect such compromises, and for end users a compromised execution is completely indistinguishable from a correct execution. Despite recent progress in secure software engineering, especially the use of formal methods to rigorously identify software vulnerabilities, it is still not possible to build bug-free software. Therefore, SGX programs will not be vulnerability free and could still be exploited. Unfortunately, traditional mitigation techniques such as ASLR will be incompatible with SGX programs because randomizing the layout of code can lead to attestation failure. As such, it is also imperative to investigate how to secure the SGX programs—especially from the enclave itself—since they cannot trust the underlying OS.

SGX can also be abused by malicious software, especially considering the fact the SGX enclave code will be invisible to the operating system. Recently, there were discussions surrounding the development of malware that can create enclaves and execute code that is undetectable and unanalyzable due to SGX blackboxing [39, 68, 69]. For example, once a system is compromised, malware can exploit Intel SGX-provided functionalities to create a botnet [29]. As such, it is unclear how the detection of malicious software running in the enclave should be performed. For example, if malware is running in the enclave, when and how does an OS kill the enclave program? Since the enclave program will appear as a giant block of code to the OS, what kind of accountability (e.g., the number of executed system calls, the number of used EPC pages, the CPU time slice) should an OS provide regarding the enclave program execution? We would like to answer these questions in this proposed research.

Research Objectives. This proposed research aims to fill the gap between what it needs from end-users, system administrators, and application developers, and what the low-level SGX instructions provide. Our approach will enable end-users and administrators to safely utilize SGX hardware and allow programmers to develop a secure enclave program without worrying about the subtle corner cases that SGX inherently contains. To be more specific, we plan to explore three research thrusts:

- **R1. Developing abstractions, libraries, and tools in support of SGX execution**. Making SGX practical requires end-to-end support from operating systems, compilers, loaders, debuggers, packaging, distributions, and programming abstractions. We will devise the best practical approach by precisely studying the pros and cons of existing alternatives, and openly develop our idea for the GNU/Linux platform. This result will not only enrich the general understanding of trusted computing and its adoption in SGX, but also contribute to the open source community.
- **R2. Securing enclave programs**. We will formulate concrete threat models of SGX programs so we can build SGX infrastructure and libraries. Specifically, we will investigate a self-defense scheme that guarantees the security of enclave programs under an adversarial OS and meanwhile investigate how to materialize traditional defense mechanisms (e.g., ASLR) for enclave programs.
- **R3. Defending against SGX malware**. Since it is impossible to trace or analyze the execution of the enclave program, malicious software can also abuse SGX to protect itself. We will explore potential accounting schemes that monitor the execution of enclave programs at different levels of granularities, thereby helping users safely maintain running enclave programs. We also plan to devise a practical policy-based scheme that allows developers to specify the runtime behavior of an enclave program.

Research Outcomes. The overall outcome we anticipate from this project is to produce systems, tools, libraries, and techniques that support and manage SGX program execution, ultimately fostering an initial movement from the open source community toward building a healthy ecosystem to secure SGX programs. As part of our effort, we will integrate our software design and artifacts into the GNU/Linux platform, and most importantly share our insights, experiment results, and software itself in top-tier systems and security venues.

Broader Impacts. This research will develop a toolchain and systems support to enable GNU/Linux application programmers to securely use SGX, while also providing a holistic solution to secure SGX programs against various attacks from a hostile OS as well as remote vulnerability exploits. It will also defend against the malicious use of SGX. These results are not only applicable to the broader scope of software development in general, but also are of interest to security, operating systems, compilers, and architectures. In addition to the broader academic impact, the proposed work is also of particular interest to the open source community as well as to the systems and security industry. We have included a collaboration letter from Intel, indicating their strong interest in this proposed research.

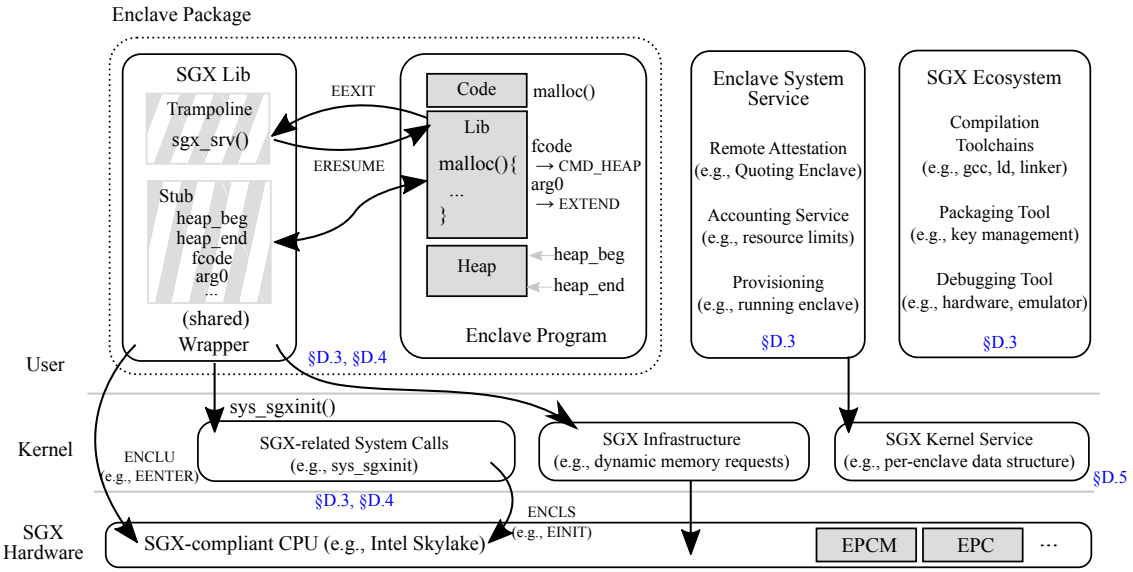

Figure 1: An Overview of the Systems and Software Support to Execute and Manage SGX Programs.

The proposed research results will be integrated into the trusted computing curriculum with lecture notes, hands-on labs, and online tutorials. These materials will be made freely available to the public. The PIs are also engaged in outreach activities, including partnering with the local (HBCU) colleges, organizing cybersecurity workshops in K-12 summer camps, organizing SGX workshops in academic conferences, and collaborating with systems and security companies.

D.2 Overview of the Proposed Research and Our Preliminary Result

This proposed research has three main goals: (1) enabling SGX program execution and management by building the corresponding systems and software support; (2) investigating techniques to secure the SGX programs running in the hardware protected enclave; and (3) defending against the malicious use of SGX.

Figure 1 shows our preliminary design of the SGX ecosystem, which consists of six major components (three located in user space and three located in kernel space) required to execute and manage enclave programs. An enclave package in user space contains the enclave program as well as the SGX runtime library that wraps the low-level SGX instructions and provides a trampoline for the enclave program. This runtime library directly interacts with the hardware, the OS kernel SGX-related system calls, and the SGX memory management infrastructures such as EPC page allocation and loading. SGX system calls will provide the system-call-level abstractions to initiate the enclave, add the EPC page, etc. There is also an enclave system service in user space that provides various functionalities regarding enclave program execution, such as provisioning, accounting, and remote attestation (e.g., launching a Quoting Enclave [39]). This enclave system service interacts with the OS kernel service. The last user space component is the SGX Ecosystem, which contains various tools for SGX program creation, packaging, and debugging. We present the design of these components in §D.3.

While SGX provides for the detection of integrity and confidentiality violations of code and data for an enclave program, we envision that there will be various attacks against SGX enclave programs. One instance is the recently demonstrated controlled side channel attack [87], in which an attacker can infer application behavior by controlling and observing the page fault of an application. There will also be other attacks from malicious OSes, such as the Iago attack [21] and Cuckoo attack [63]. Meanwhile, enclave programs may still contain conventional memory corruption bugs (e.g., buffer overflow) that attackers can exploit. We will explore security mechanisms to ensure the safety of enclave programs by strictly following two principles: 1)

security by construction and 2) the principle of least privilege. The details of our proposed research in this direction are presented in §D.4.

SGX is designed to prevent privileged software from analyzing the enclave program. Malicious software will likely use SGX to defeat malware analysis [29, 69]. Meanwhile, when SGX is introduced to the desktop ecosystem, keyloggers and screen readers will trivially sniff enclave programs' secrets. Therefore, in this proposed research we would also like to investigate how to detect the malicious use of SGX. We will look into both anomaly detection and misuse detection, and design new threat models regarding SGX malware. We will also develop support for the OS kernels to incorporate other possible trusted software or hardware components to secure I/O in a desktop environment. We provide our research plan for defending against malicious use of SGX in §D.5.

Preliminary Result. Both PI Kim and PI Lin are among the pioneers of SGX research. In the past a few years, PI Kim has led the efforts to develop OpenSGX (`https://github.com/sslab-gatech/opensgx`), which is an open source emulator of the new SGX instruction set and allows researchers to evaluate their prototypes when using the SGX platform [1]. As of today, OpenSGX has been used by more than 20 research groups worldwide including the group led by PI Lin. OpenSGX has built a strong foundation in developing systems and software support for SGX program execution.

PI Lin also has been working on investigating SGX for new security applications. In particular, PI Lin is among the first few research groups to have full access to the SGX platform from Intel, and he has led efforts to investigate the analysis of SGX enclave programs, especially for malware analysis applications. He also has looked into securing cryptographic keys using SGX, as well as protecting other secrets such as gaming data in enclave programs.

PI Kim and PI Lin also have been collaborating to develop SGX use cases. One such a use case is to secure a TOR network node by using OpenSGX to protect cryptographic keys [2]. Another is to study the system-level behaviors of enclave programs.

More importantly, the knowledge and early experiences from building the basic development platform for the SGX emulator and designing their applications have equipped the PIs with a deep understanding of SGX's advantages and disadvantages. Based on these experiences and strong support from Intel, we believe our team can advance the current state of the art in secure SGX execution, management, and defense.

D.3 R1: Developing Systems and Software Support for SGX Program Execution

Proper use of SGX requires support from the OS kernel with corresponding system calls and management services (§D.3.1), user-level runtime libraries and service routines (§D.3.2), and enclave program development toolchains (§D.3.3). In this section, we describe the justification of our design decisions with technical challenges and our initial ideas to overcome them in the proposed work.

D.3.1 R1.1 Operating Systems Support

Since SGX includes a number of privileged instructions (e.g., `EINIT`), they must be executed by OS kernels. This implies that an operating system, an untrusted entity with respect to SGX, must be involved to provide a service (e.g., through system calls) to use these instructions.

These privileged instructions are necessary for an OS to initiate an enclave, allocate enclave pages (EPC pages), and maintain the address mapping (e.g., fetching EPC information in the BIOS and maintaining the related page table). We also believe the OS needs to support quality of service (QoS) for enclave programs. For example, how can we allow users to kill a suspicious enclave program when it is so privileged that the user cannot see its execution? The OS should provide an interface for users to enforce policy and keep track of

[1] The details describing how OpenSGX works are currently under submission.

[2] The academic paper describing the details of protecting a TOR node is also under submission.

Instruction	Description

`int sys_init_enclave(`
`  void *base_address,`
`  unsigned int n_of_pages,`
`  tcs_t tcs,`
`  sigstruct_t *sig,`
`  einittoken_t *token)`
	Allocate, add, measure EPC pages, and initialize OS-specific structures
	Starting address of code/data pages, a linear address.
	The number of total pages to be loaded.
	Thread control structure address used for entering enclave, a linear address
	Information about the enclave from the enclave signer
	Token for verifying that the enclave is permitted to launch

	Related leaf commands: `ECREATE`, `EADD`, `EEXTEND`, `EINIT`
`void sys_terminate_enclave(` `  int keid)`	Terminate an enclave Enclave ID (maintained inside the kernel)
	Related leaf commands: `EREMOVE`
`epc_t sys_add_epc(` `  int keid)`	Allocate a new EPC page to the running enclave. Enclave ID (maintained inside the kernel)
	Related leaf commands: `EAUG`

`int sys_stat_enclave(`
`  int keid,`
`  enclave_info_t *stat)`
	Obtains the enclave stats: such as eid, #encls, #enclu calls, allocated stack/heap, perf etc.
	Enclave ID (maintained inside the kernel)
	Container of stat information of enclave

	Related leaf commands: `N/A`

Table 3: List of potential system calls to support, execute, and manage enclave programs in commodity operating systems.

each enclave instance. In addition to these management issues, a set of other features should be implemented for proper functionality, such as dynamic EPC page allocation and expansion. Our team first devised a list of four system calls to execute, terminate, and maintain enclave programs, as summarized in Table 3.

(A) **Bootstrapping.** Not all memory pages can be used as EPC. When booting the kernel, the OS needs to know the mapping of enclave pages; a user can configure how many EPC pages can be allocated via BIOS. Upon booting, the OS kernel obtains a contiguous chunk of EPC and its physical address. Then, the OS can use the EPC region to initialize an enclave.

(B) **Enclave initialization.** Intel SGX provides four instructions related to enclave initialization: `ECREATE` (create an enclave), `EADD` (add a page), `EEXTEND` (extend EPC page measurement), and `EINIT` (initialize an enclave). Since system call numbers have a limited maximum value, we should introduce as small a number of system calls as possible. Therefore, we propose to design one system call, namely, `sys_init_enclave()`, to initiate, extend, and measure an enclave program that requires privileged SGX instructions.

(C) **Dynamic EPC page allocation.** The Intel SGX revision 2 [11, 43] introduces a mechanism to dynamically expand enclave memory by using `EAUG` and `EACCEPT`. Based on these two instructions, we propose to provide `sys_add_epc()` to dynamically allocate additional EPC pages for an enclave that requires more memory. When an enclave needs a new EPC page, the OS allocates a free EPC page via `EAUG`. Then, the enclave should invoke `EACCEPT` to accept the new page into its own enclave region.

Research Challenge 1. *A hostile OS can attack the enclave program by controlling EPC page allocation and deallocation. How can we defeat this attack?*

We propose to perform an integrity check upon the execution of the `EACCEPT` instruction (recall we will provide wrappers to all the user-level SGX instructions), to thwart such an attack. The details of this integrity check are presented in §D.4.1.

(D) **Enclave Program Execution Accounting.** Finally, the operating system should be in charge of authorization, fairness, and execution of requested enclave programs to fully take advantage of SGX features. For instance, to support multiple enclaves concurrently, the OS kernel needs to maintain a per-enclave structure that describes the execution context of each enclave. This would include the enclave ID and

API	Description
`void sgx_enter(tcs_t tcs, void (*aep)())`	`EENTER` wrapper
`void sgx_resume(tcs_t tcs, void (*aep)())`	`ERESUME` wrapper
`void launch_quoting_enclave(void)`	Launch quoting enclave
`void sgx_exit(void *addr)`	`EEXIT` wrapper
`void sgx_remote(const struct sockaddr *target_addr, socklen_t addrlen)`	Request remote attestation
`void sgx_getkey(keyrequest_t keyreq, void *key)`	`EGETKEY` wrapper
`void sgx_getreport(targetinfo_t info, reportdata_t data, report_t *report)`	`EREPORT` wrapper
`int sgx_enclave_read(void *buf, int len)`	Read from host
`int sgx_enclave_write(void *buf, int len)`	Write to host
`void *sgx_memcpy(void *dest, const void *src, size_t size)`	Memory copy
`void *sgx_memmove(void *dest, const void *src, size_t size)`	Memory copy
`void sgx_memset(void *ptr, int value, size_t num)`	Memory set to the specified value
`int sgx_memcmp(const void *ptr1, const void *ptr2, size_t num)`	Memory comparison
`size_t sgx_strlen(const char *string)`	Get string length
`int sgx_strcmp(const char *p1, const char *p2)`	String comparison
`int sgx_printf(const char *format, ...)`	Write formatted data to standard out

Table 4: List of APIs to be implemented in `sgxlib`.

the contents of `TCS` and the stack size, similar to the `task_struct` in Linux. The structure also needs to contain debugging and accounting information (e.g., the number EPC page allocated, the CPU time slice executed, etc.).

Research Challenge 2. *What is the best way to provide contextual information about enclave executions to users? How effective is the information in determining abnormal behavior of enclave executions, and how efficiently can our infrastructure keep track of them?*

As such, we propose to design a `sys_stat_enclave()` system call for this purpose. When an enclave is created, we need to keep track of the new enclave by assigning an identifier (`keid`) in the kernel and a descriptor. For the given `keid`, the enclave descriptor collects stat/profiling information, including statistics and enclave-specific metadata (e.g., `SECS` and `TCS`). A utility application can later query the collected accounting information through `sys_stat_enclave()`.

D.3.2 R1.2 Runtime Libraries and User Level Enclave System Service Support

We also anticipate there is a need for a system runtime library that provides necessary functions for SGX programmers to use and execute enclave programs. Therefore, we propose to design such a library (called `sgxlib`) with the goal of (1) facilitating enclave programming and (2) minimizing the attack surface between the enclave and its *potentially malicious* host process. Table 4 lists APIs that need to be implemented by `sgxlib`. Below we describe how we should design the `sgxlib` and its security considerations.

Research Challenge 3. *Standard `C` libraries, such as `glibc`, are frequently used by normal `C` programs. However, using standard `C` libraries inside an enclave raises two concerns: (1) any function call that relies on OS features or resources will break the execution of enclave programs, and (2) enabling such functions opens a new attack surface (e.g., a malicious host can return a crafted input to the enclave).*

We propose to implement a number of custom library functions that have a similar counterpart in the standard library, but we add a `sgx_` prefix to distinguish the two (e.g., `sgx_memmove()` for `memmove()`). While this is an engineering challenge, we plan to utilize an automatic source code instrumentation tool (e.g., LLVM [50] or CIL [59]) to speed up our development.

Research Challenge 4. *Although an enclave can legitimately access memory shared outside the enclave, this is not a recommended practice since a malicious host or operating system can potentially modify non-enclave memory. Consequently, how do we handle non-enclave memory for an enclave program?*

We propose a stricter form of communication protocols by using shared code and data memory—we call them *trampoline* and *stub*, respectively. The use of trampoline and stub defines a narrow interface to the enclave that is tractable for enforcing the associated security properties.

More specifically, our stricter communication is one-way and entirely driven by the requesting enclave. For example, to request a socket for networking (see Figure 2), the enclave first sets up the input parameters in *stub* (e.g., sets `fcode` to `FSOCKET` in Figure 2), and then invokes a predefined handler, *trampoline*, by exiting *enclave mode* (i.e., by invoking `EEXIT`). Once the OS processes the enclave request, it stores the result or return values to `stub`, and enters *enclave mode* by invoking `ERESUME`. After transferring the program's control back to the known location inside the enclave, the enclave program can, finally, obtain the returned value (e.g., socket via `in_arg0` in *stub*).

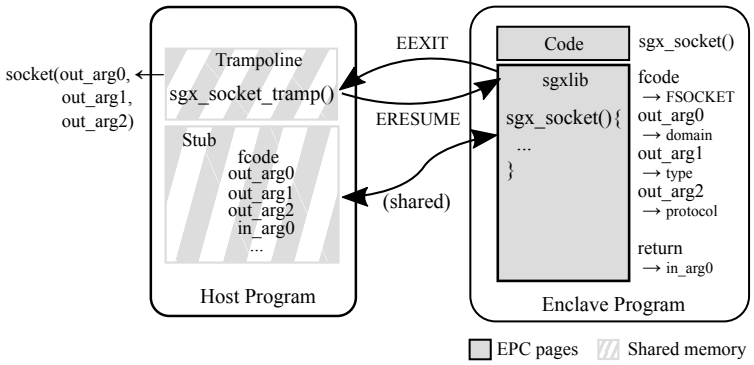

Figure 2: Interface defined for communicating with the non-enclave host program that performs the delegated calls to the operating system. In this figure, an `sgxlib` function, `sgx_socket()`, running inside the enclave, requests a socket system call via *trampoline* and *stub*, which are pre-negotiated between the enclave and its wrapper when packaged together.

Dynamic Memory Allocation. In addition to the runtime libraries, we also need to design and implement other user libraries necessary for the execution of enclave programs. One such a library is for the handling of dynamic memory allocation. Although it is permissible for an enclave program to use dynamically allocated user memory, it can break the enclave isolation feature. To avoid this, we propose a customized dynamic memory allocation API, `sgx_malloc()`, that behaves similarly to glibc `malloc()` [86], but that only allocates memory from the enclave heap (pre-allocated EPC pages, see Figure 3). `sgx_malloc()` manages the enclave heap by maintaining heap pointers, which are initially set to the heap with the aid of the OS during the first initial `sgx_malloc()` call. When the pre-allocated heap area becomes full, `sgx_malloc()` leverages dynamic EPC page allocation (via `sys_add_epc()`) to extend the enclave heap. With `EAUG`/`EACCEPT`, dynamic EPC page allocation ensures that only a zero-filled EPC page, with an associated pending bit of `EPCM`, is added to the enclave that invoked `EACCEPT`. Since the pending bit can be switched only by executing `EAUG`/`EACCEPT`, a malicious OS cannot trick the enclave into adding another EPC page.

Remote Attestation. We also need to support remote attestation, a crucial execution step in SGX. To this end, we propose an API `sgx_remote()` with which programmers can generate a remote attestation request in an enclave program. It uses `sgx_getkey()` and `sgx_getreport()` to get a report key and create a report. By specifying the socket information of a target enclave, a challenger can issue a remote attestation to check (1) the correctness of the target program (based on the hash of EPC contents) and (2) whether it is actually running inside an enclave on the SGX-enabled platform (MAC with report key). To launch and serve a special enclave (called a "quoting enclave") that verifies a target enclave through intra-attestation, we also need to provide `launch_quoting_enclave()`. The overall procedure of remote attestation will be implemented based on the SGX specification [10].

D.3.3 R1.3 SGX Toolchain and Ecosystem Support

The final systems component for SGX is the toolchain for application development. Currently, there is no official document of the SGX application binary interface. The toolchain we propose includes a compiler, loader, debugger, and a new packaging tool packer.

Compiler. We will not directly modify any existing code base of mainstream compilers such as `gcc` or `clang`. Instead, we will use them to compile an enclave program. Because we will provide wrappers to SGX instructions, as long as we eventually link to these wrappers (whose object code is produced by the Intel assembler) we do not have to modify the compilers. However, we do have to take special care while compiling an enclave program. Specifically, one key feature of an SGX binary is that it should be easily relocated to EPC. According to the SGX specification, the `EADD` instruction loads code and data into the EPC by direct memory copying, which implicitly assumes that developers take care of program relocation by themselves. To ease developers' efforts in handling program relocation, we propose a build script to automatically adjust compilation options to make enclave code and data easily relocatable at runtime.

Loader. An SGX binary loader needs to determine the memory layout of code, data, stack, heap, and necessary data structures on the EPC region during initialization of an enclave. Similar to 'loader' in the OS, our SGX binary loader should obtain the information of code and data sections (i.e., offset and size of `.enc_text` and `.enc_data` sections) and the program base address from corresponding ELF files. The required enclave size and the memory layout are determined based on code and data size, memory configuration, and other necessary data structures (see Figure 3). Then, our SGX binary loader forwards the memory layout information to the `sys_init_enclave` system call to initiate the enclave.

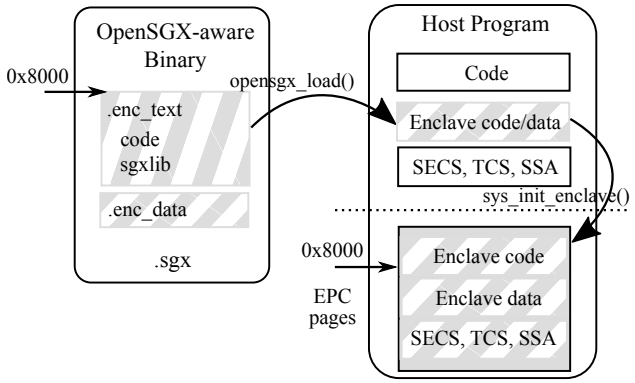

Figure 3: Loading process performed by our loader. First, `.enc_text` and `.enc_data` sections are loaded in to host memory. The SGX loader then forwards two sections along with stack, heap, and other necessary data structures to EPC via `sys_init_enclave()`.

Research Challenge 5. *SGX requires that application code and data be placed on an enclave page cache (EPC), a reserved, encrypted area of memory, and execution must stay within EPC. When executing a binary on EPC, an SGX instruction can allow one to copy a normal page onto an EPC page. Therefore, we have to enable our dynamic loader to provision the code, data, and stack sections on EPC and to also handle the EPC relocation appropriately. This is non-trivial. Meanwhile, such dynamic code loading into the enclave also introduces obstacles for remote attestation. We have to address these additional challenges.*

Debugger. To debug an enclave program, Intel SGX CPUs have hardware support for a single step of an enclave program. Therefore, we need to make `gdb` (our base platform) aware of this feature. To this end, in addition to enabling the single step inside an enclave, we propose to modify `gdb` and introduce four new commands. These are `info epc`, `info epcm`, and `info secs` to examine EPC-related data structure, and `list enclaves` to list all the active enclaves (and their contexts) with corresponding `eid`.

SGX Binary Packer. Once the binary is compiled and linked with our library, we also need to generate other help files such as the hash of the code, which will be used for attestation measurement. To this end, we need to develop an SGX binary packing tool. It will generate a configuration file (a .conf file) that contains the program measurement (a hash value), a signature signed by a specified RSA key, and other enclave properties that are required to validate the program during enclave initialization.

We also expect that it will be helpful to specify the system calls (e.g., the system call name, parameters, etc.) that will be used by the enclave program in the configuration file. Much like smartphone apps that ask for permissions [32, 33, 45], if the enclave can explicitly ask for permission to execute system calls, this will allow the OS to enforce security policies. This is especially helpful when considering that enclave programs

Type	Interface	Attack surfaces	In-enclave usage/check
MEM	sgx_malloc() → <out>addr	1) incorrect pointers, 2) incorrect EPC add.	EACCEPT verifies the status of a new EPC
MEM	sgx_free() → N/A	1) not freed (used later for use-after-free)	sgx_free() fills a freed chunk with zero
DBG	sgx_puts() → N/A	1) ignored output	No general way to prevent wo/ trusted I/O
TIME	sgx_time() → <out>time	1) arbitrary time	Validate time from the NTP server
RAND	sgx_rand() → <out>rand	1) arbitrary value	Relying on rdrand inst
IO	sgx_write() → <out>len	1) arbitrary reported len	No general way to prevent wo/ trusted I/O
IO	sgx_read(<out>*buf) → <out>len	1) crafted buf, 2) incorrect len	Encrypted message with integrity checking
IO	sgx_close() → N/A	1) not closed	Never reuse fd (monotonically increasing)
NET	sgx_socket() → <out>fd	1) non-closed fd, 2) incorrect fd	Relying on packet encryption
NET	sgx_send() → N/A	1) ignored	Relying on packet encryption
NET	sgx_recv(<out>*buf) → <out>len	1) crafted buf, 2) incorrect len	Relying on packet encryption
NET	sgx_accept() → <out>fd	1) pre-allocated fd, 2) arbitrary number	Relying on packet encryption
NET	sgx_bind() → N/A	1) failed binding	Stop-on-failure
NET	sgx_listen() → N/A	1) failed listen	Stop-on-failure
NET	sgx_connect() → <out>err	1) failed connection	Stop-on-failure

Table 5: Consideration of Iago attack in primitive sgxlib functions that are to be implemented by using the shared trampoline between OS and enclave programs. Note that Intel SGX does not consider denial-of-service attacks (e.g., stopping enclave execution) nor strong privacy (e.g., where to talk to).

can be attacked, and that there could be malware running in an enclave. We plan to develop automatic techniques to derive the system call specifications from the enclave programs and enforce them with system call level sandboxing [36, 66, 81]. More details are discussed in §D.5.

D.4 R2. Securing SGX Enclave Programs

Research Challenge 6. *An enclave program inevitably must interact with the underlying OS to request the services it provides (e.g., system calls), as well as the resources it manages (e.g., virtual pages, random numbers). This makes the enclave programs particularly vulnerable to a number of layer-below attacks. In particular, enclave programs rely on the support of an underlying OS, but the security model of the enclave programs excludes the OS from the TCB. Such an unconventional dependency makes various attack vectors, e.g., Iago and side-channel attacks, which are often considered impractical in a traditional setting, immediate and practical, especially in a cloud environment.*

In this research thrust, we would like to revisit how to defeat these attacks. Within the context of SGX, we want to defeat Iago attacks (§D.4.1) and controlled side channels (§D.4.2), and also achieve secure I/O under a hostile environment (§D.4.3). In addition, enclave programs can contain exploitable vulnerabilities such as buffer overflows, and typical threats and attack vectors still exist. We would like to investigate techniques to mitigate typical vulnerability exploitation as well (§D.4.4).

D.4.1 R2.1 Defeating Iago Attacks

To provide enclaves the ability to communicate with the underlying OS, it is unavoidable to introduce additional attack surfaces. Iago attacks [21], in which the operating system or the underlying hypervisor (e.g., [35, 46]) manipulates the return values and actions performed by the system call, are introduced when enclaves interact with the OS. However, by taking precautions inside an enclave, potential attacks can be prevented. In our research, we intend to prevent Iago attacks by systematically protecting the enclave from malicious system call results using the enclave library sgxlib. Specifically, we propose to defend against Iago attacks in three broad aspects: dynamic memory allocation, network and I/O, and non-determinism/resources. Note that Intel SGX does not prevent denial-of-service attacks or guarantee strong privacy (e.g., IP address), but provides strong isolation and confidentiality. With this in mind, as outlined in Table 5, we inspect the potential attack surface in communication between the enclave program and underlying kernel and discuss self-defenses from the enclave program itself in each category of attack.

- **Memory-related operations (marked `MEM`):** Since SGX revision 2, enclave programs can dynamically request EPC pages at runtime, which would provide the operating system with a potentially powerful attack vector. However, Intel SGX takes this into account and provides the `EACCEPT` instruction, which performs basic validation on newly assigned EPC pages (e.g., non-enclave pages or pre-allocated EPC pages). This thwarts a major source of memory-related attacks. We would like to investigate this feature and use it to defeat memory-related Iago attacks.
- **Network and I/O services (marked `IO`, `NET`):** Two principles are considered to prevent network- and I/O-related attacks: *encryption* and *fail-stop model*. Since a malicious OS may attempt to read or manipulate I/O from an application, to guarantee the confidentiality of packets, we believe enclave programs should encrypt all outgoing packets and validate the integrity and authenticity of all incoming packets. Upon any failure, the enclave stops its execution, which can dramatically reduce the potential attack surface in handling all errors and corner cases. We will also go into more details below about securing other I/O such as keystrokes and screen outputs in §D.4.3.
- **Non-determinism and resources (marked `DBG`, `TIME`, `RAND`):** Enclave programs often need time and randomness to provide rich experiences to users, but such data cannot simply be requested from the OS. To prevent Iago attacks, we will investigate the various ways this data can be obtained. For instance, we can fetch time values from trusted parties(e.g., an encrypted packet from known NTP servers) or randomness from a trusted CPU (e.g., the `rdrand` instruction).

D.4.2 R2.2 Defending Against Controlled Side Channel Attacks

Given the awkward TCB of enclave programs (i.e., no trust of the underlying OS), the OS has the capability to infer enclave application behavior through controlled channels. For instance, an OS can control the CPU execution of an application (e.g., the scheduling of for multi-threaded applications, which may lead to different behavior if there are dependencies among the threads), as well as the memory usage of an application (e.g., a page fault). Most recently, Xu et. al. [87] showed that analyzing an application in advance can enable an attacker to later reconstruct with surprising accuracy the data and control flow of an application by analyzing only its patterns of page faults. Since enclaves must still request memory, any applications not specifically designed to avoid leaking information in this manner may be vulnerable to revealing private information even if the entire application is placed in an enclave. To defeat such controlled side channel attacks, we plan to investigate defense from within the enclave program itself. We believe there are two basic strategies to self-defend against the controlled page fault side channel:

- **Enclave code rewriting**. Since controlled side channel attacks try to infer a user's input, it is possible to rewrite binary code such that the memory access pattern does not depend on sensitive data. Prior studies have demonstrated that with a special compiler, it is possible to mitigate the timing-based side channel [25]. We plan to use our experience in binary code rewriting (e.g., [84, 85, 89]) to defeat this attack.
- **Page fault noise injection**. The other strategy is to inject page fault noise. For instance, we can control the binary code to trigger a large volume of page fault noise that is not related to user input. We can also randomize the binary code at runtime to trigger unobserved page faults. We plan to investigate these alternatives and test their effectiveness.

There are also other potential controlled side channel attacks, e.g., the patterns in an application's system calls to the operating system. Since we cannot prevent enclave programs from using system calls, injecting system call noise may be one option to mitigate controlled side channel attacks. We plan to investigate this defense as well.

D.4.3 R2.3 Securing Keystrokes and Screen I/O

Research Challenge 7. *Intel SGX is an ideal model for the cloud, as it has a very restricted set of the I/O channels (usually just network communication). To use Intel SGX in a desktop-like environment, it is essential to es-*

tablish a secure channel between users and the enclave program via the keyboard and monitor. But such I/O is out of the scope of the TCB. How can we create a secure channel between the enclave programs and end users?

Since enclaves themselves can only run in ring 3, they are not well adapted to protecting I/O drivers. For instance, the keyboard driver can be completely malicious (i.e., a keylogger), and the screen output buffer can be intercepted by a malicious video card driver, etc. However, securing these inputs and outputs is paramount, especially when SGX is pushed to desktop applications, such as those used for DRM, online gaming, online shopping, online banking, etc.

Securing these forms of I/O requires that the endpoints be trusted or be able to attest that they can be trusted. From a client/server perspective, I/O is mainly viewed from the perspective of network communications, whose trust has already been addressed via encryption and remote attestation. Keyboards and screen buffers must also provide trust. One way this can be addressed is with special trusted hardware devices. There are a few commodity hardware devices already available on the market: Intel Protected Audio and Video (PAVP) [41] and Intel Identity Protection Technology (Intel IPT) [16]. PAVP [41] provides an encrypted data path and hardware accelerated decoding for high-definition video and audio playback. Intel IPT [16] provides a trustworthy display, called the Protected Transaction Display (PTD), so the display to users can be protected from malicious applications or compromised operating systems. We will investigate the use of these trusted hardware to build a secure channel between end-users and the enclave programs.

D.4.4 R2.4 Mitigating Vulnerabilities in Enclave Programs

Research Challenge 8. *SGX does not automatically secure vulnerable enclave programs; typical threats and attack vectors such as buffer overflows still remain. In fact, the compromise of an enclave program becomes more critical because SGX makes it impossible to analyze or even detect such compromises, and for end users, a compromised execution is completely indistinguishable from the correct execution (when an attacker also hides the system call footprints).*

Despite recent progress in model checking and formal methods [15, 23, 44, 47, 51], we cannot produce vulnerability-free software. There will be still an eternal war of memory errors between attackers and defenders [78] that applies to enclave programs as well. Even worse, previous OS-level defenses will become ineffective because fine-grained program behavior is invisible to the OS. While address space layout randomization (ASLR) may appear to mitigate this class of vulnerability, randomizing the layout of program code will lead to attestation failure for an enclave program. Therefore, we must search for new or revisit existing techniques to secure an enclave program from the enclave itself. Since we will not trust the underlying OS, we must use the self-defense mechanism in the fight against memory exploits. There are two practical strategies in this direction:

- **Control flow integrity (CFI)**. Since all control flow hijacks lead to a violation of control flow, CFI [8] or more generally various software guards (e.g., [9, 14, 17, 18, 26, 31, 71]) are proposed to defend against these attacks. While initial CFI approaches required access to the debugging symbols of a program, recent progress has made CFI largely practical. For instance, CCFIR [90] rewrites the binary with the relocation information from the COTS binary and enforces a practical CFI model. BinCFI [92] proposes a number of disassembling and rewriting strategies and can successfully build a CFI model for large COTS binaries. In this proposed research, we would like to revisit these practical CFI systems with the new constraints of handling cross enclave control flow transfers and also self-detecting violations by the enclave program itself.
- **Binary code randomization**. Most control flow hijacks need to know to where control flow transfers, and reuse this code to construct the exploit. As such, randomizing the layout of program code can significantly mitigate these attacks. ASLR has been shown to be effective in this direction [12, 75, 79]. However, remote attestation will fail if we directly attest the randomized code in the enclave. Instead, we need to derandomize the code and compute its attestation hash value. We plan to address this

challenge by developing a self-randomized ASLR scheme inside the enclave. Our prior effort has demonstrated that we can randomize instruction addresses every time we load a program, defeating code reuse attacks [85]. There are also several other efforts along this (re-)randomization direction (e.g., [13, 27, 28, 37, 38, 62]). We plan to integrate these re-randomization techniques while still addressing the attestation issue for enclave programs.

D.5 R3. Defending Against Malicious Use of SGX

Research Challenge 9. *We have discussed that an SGX program can be compromised because of exploitable vulnerabilities. There are also cases in which the SGX programs themselves can be entirely malicious. For instance, malware can now be executed inside an enclave that is undetectable and unanalyzable due to SGX blackboxing [69]. This will be especially problematic when SGX is widely available in desktop computers. On the other hand, while SGX is designed to defend attacks such as cloud providers in leaking or tampering with the sensitive code and data, cloud providers also have to make sure that there is no malicious SGX program that causes damages to their infrastructures. Therefore, it is also critical to study how to defeat malicious use of SGX.*

We believe there are two strategies to defeat the malicious use of SGX. One is to detect malicious software based on certain signatures (§D.5.1), and the other is to enforce the access control of enclave program execution based on explicit policies (§D.5.2).

D.5.1 R3.1 Detecting Malware Execution in the Enclave

Traditional intrusion detection systems utilize signature matching to detect the presence of malicious software, as well as anomaly-based detection using benign behavior profiles [19, 49, 60, 73]. We plan to design similar techniques to extract signatures of enclave program execution.

We cannot look at any control flow related information inside the enclave. Instead, we can only look at those operating system observable behaviors, such as system calls (including arguments and return values) [48] and their sequences [40, 83] or graphs [61]. We can also look at unique features in SGX program execution, such as the use of EPC pages, CPU time slice, the frequencies of page fault, etc. Therefore, we plan to investigate combining system call and SGX execution features to build a profile for malicious enclave programs (if we have a priori knowledge of their existence) as well as profiles for benign enclave programs.

D.5.2 R3.2 Preventing Malware Execution in the Enclave

The execution of an enclave program provides a perfect opportunity for the OS to enforce an execution policy (such as only allowing the execution of a system call that is registered). While it is challenging to acquire such a model through program analysis, we can ask programmers to explicitly register or develop tools (e.g., as in [20, 24, 52]) to help SGX programmers in expressing these policies. Considering there is no legacy SGX code at this moment, we believe this is a practical way of executing the SGX program. Each enclave program will have an associated system call execution policy file that specifies which system call (perhaps including its parameters) the enclave program will execute. The OS will enforce the execution of these system calls [34, 67]. In fact, the recent program execution model in mobile apps has already adopted a similar approach, in which an app has to explicitly express its resource use [32, 33, 45].

More broadly, we can also adopt the existing code producer and verifier execution model (such as the proof-carrying-code [58]) for SGX programs. For instance, we can restrict how an SGX program is produced (e.g., by a special compiler) and signed, and then verify certain properties of the code before executing it in the enclave. In fact, such an approach has been used in several existing systems. For example, Google's Native Client (NaCl) [88] requires a special compiler to modify the client programs at the source level. Microsoft's CFI [8] and XFI [31] requires code-producers to supply a *program database* (PDB) file with their released binaries. PittSFIeld [54] and SASI [30] require code-producers to provide `gcc`-produced assembly code.

Task	Success Metrics and Evaluation Plan	Time Line (RA-month)							
		Y1		Y2		Y3		Y4	
		G	U	G	U	G	U	G	U
R1.1	Linux kernel patches that support enclave program execution in the most recent kernel will be delivered and evaluated by the open source community.	12	12						
R1.2	A user level runtime library (with a lot of `glibc` code refactoring) and enclave system services (e.g., remote attestation) will be developed and also evaluated by the open source enclave program developers.	12	12						
R1.3	The SGX toolchain and ecosystem (e.g., packaging) support will be developed including compiler, loader, debugger, and binary packer. This will also be tested by open source enclave program developers. Meanwhile, we will also refactor the existing SPEC2006 benchmark and execute them in our platform to evaluate their performance overhead.		12	6	12				
R2.1	An `sgxlib` will be developed with special care of defending Iago attacks. This library will be tested with a variety of layer below attacks from OS kernels and hypervisors. We will use the benchmarks in [21, 35, 46] for the evaluation.			12	12				
R2.2	An enclave binary code rewriting, and a page fault noise injection scheme will be developed, and evaluated with benchmarks in [87] to defeat the controlled side channel attacks.			6	12	6			
R2.3	A scheme using trusted hardware to secure the keystrokes and screen I/O will be developed, and tested with keyloggers and screen buffer readers we will develop.					6	12		
R2.4	A binary level control flow integrity scheme, and a binary code randomization scheme will be developed and tested with the vulnerable enclave programs.					12	12		
R3.1	An enclave program execution profile will be developed, and then an intrusion detection model will be built based on the profile. Malicious or benign enclave program will be developed and tested with our detection model.							12	12
R3.2	A system call sandbox for enclave program execution will be developed, along with a system call policy derivation tool and enclave code verifier. They will also be tested with real enclave programs such as the SPEC2006 benchmark we refactored in R1.3.							12	12

Table 6: Tentative project time line and the detailed evaluation plan (Note that Yi denotes the i-th Year, G stands for Georgia Tech, and U stands for UT Dallas).

Therefore, when designing SGX malware execution prevention system, we would also like to investigate the model of restricting how the SGX code is produced, and then we can decline the execution of the code that does not pass the code verifiers. Such an approach will also bring many other advantages. For instance, we may be able to verify whether an enclave program contains incorrect logic that can expose secret key to memory outside the enclave during the verification process.

D.6 Broader Impacts: Education and Outreach

Curriculum Development. The systems security courses taught by both PIs will benefit greatly from the proposed research. In particular, PI Kim introduced the concept and recent research progress surrounding SGX into the course "Building Secure Systems (CS 8803)". In fall 2014, a group of graduate students initiated OpenSGX as part of their course project. The team leader, Soham Desai, joined the Intel Security Team after his graduation. PI Kim continues to improve current lab materials (limited to SGX itself) by introducing new security applications as part of this proposed work.

PI Lin also has been teaching a systems security class (CS 6332) since 2012. Given the significance of SGX, PI Lin has already developed course materials related to SGX. These include basic concepts to real implementation and hands-on projects, and were taught in fall 2015. The outcome of this research, especially the programming abstractions and toolchains, will serve as the enabling vehicle for student hands-on projects. PI Lin plans to have an SGX project using the OpenSGX platform developed by PI Kim in his fall 2015 class.

Outreach to undergraduates, women, and minorities. The University System of Georgia (USG) recently started an ambitious effort to increase the cyber security workforce across all units of the USG system (29 campuses). These include multiple HBCU colleges, including Spelman and Morehouse colleges, and research universities. Georgia Tech is expected to play a leading role in developing effective programs that deploy cyber security content to a diverse set of programs. Dr. Kim will lead the systems security content development in this area, including curriculum development and an "educating the educators" program to

maximize the number of impacted students. As part of this work, the course material and labs regarding SGX will be developed and introduced to the local security community.

PI Lin has a history of successfully mentoring undergraduate students in research. One undergraduate advisee (Garrett Greenwood) is a co-author of an NDSS 2014 paper, and another undergraduate student, Devin Willey, won the first place prize in the UT Dallas CS Department's First Annual Undergraduate Research Expo in 2015. Dr. Lin is also currently supervising one female Ph.D. student. Finally, Dr. Lin plans to organize cybersecurity workshops in K-12 summer camps, describing the recent advances in securing the cloud including SGX.

Outreach to the research and development community. The PIs plan to organize an SGX centralized workshop or tutorials at conferences such as ACM CCS. Given the importance of cloud computing today, there is a pressing need to adopt techniques such as SGX for secure cloud computing. Organizing an annual SGX workshop can bring together both academic researchers and industry practitioners to share their perspectives on SGX development, and update participants with the most recent progress in this field.

PI Kim is an active contributor to the open source community; not only did he initiate and lead the OpenSGX project, but he also works closely with practitioners. For example, the security group PI Kim leads was invited to the 2015 Linux Plumbers Conference to present another open source project that enables rebootless kernel updates. To foster successful deployment and contribution to the community, the developed design, technical reports, and software artifacts developed through this proposed work will be publicly available.

D.7 Evaluation and Management Plan

We anticipate this project will require four years to complete. Table 6 outlines a tentative schedule of anticipated yearly foci for each specific research task, the corresponding evaluation plan, and the dedicated months in terms of the graduated research assistant (RA) from each institute. The details on how the two PIs collaborate and manage the project are presented in the attached **Collaboration Plan**.

D.8 Results From Prior NSF Support

Taesoo Kim: Dr. Kim is the PI of *"SaTC-EDU: EAGER: Big Data and Security: Educating The Next-Generation Security Analysts"* (NSF DGE 1500084, 06/2016-05/2018, $300,000). **Intellectual merit**: the identification of a set of basic principles and effective techniques of applying data analysis to security education, and the contribution to research and practice in data analysis and security. **Broader impacts**: newly designed, developed, and evaluated publicly available resources such as lab materials for teaching data-driven security analysts, as well as the training of better technical quality of security analysts.

Zhiqiang Lin: Dr. Lin is the PI on *"CAREER: A Dual-VM Binary Code Reuse Based Framework for Automated Virtual Machine Introspection"* (NSF CNS 1453011, 09/2015-08/2020, $535,054). **Intellectual merit**: the development of a novel dual-VM binary code reuse-based framework to automatically bridge the semantic gap for virtual machine introspection. **Broader impacts**: better and automated protection of computer systems against advanced malware attacks, as well as training of new security students and researchers. Lin is co-PI on *"UTD SFS Renewal: Growing a Cybersecurity Community through SFS Scholarship Program at UTD"* (NSF DGE 1433753, 09/2014-08/2019, $3.9M). **Intellectual merit**: Recruitment and training of students in our cyber security education programs and the development of a new concentration track in this direction. **Broader impacts**: successful placement of students in federal government, and various outreach activities to students in Texas. Lin is also co-PI on *"Virtual Laboratory and Curriculum Development for Secure Mobile Computing"* (NSF DGE 1516425, 08/2015-07/2017, $299,634). **Intellectual merit**: developing an Android based virtual lab for education in malware analysis and enhancing educational experiences for students via several programming projects on a mobile platform. **Broader impacts**: the proposed virtual lab is a valuable tool for mobile apps developers and researchers, and is accessible to anyone.

E References Cited

[1] M. Abadi, M. Budiu, Ú. Erlingsson, and J. Ligatti. Control-flow integrity principles, implementations, and applications. *ACM Transactions on Information and System Security*, 13(1), 2009.

[2] P. Akritidis, C. Cadar, C. Raiciu, M. Costa, and M. Castro. Preventing memory error exploits with WIT. In *Proceedings of the 29th IEEE Symposium on Security and Privacy (Oakland)*, pages 263–277, Oakland, CA, May 2008.

[3] I. Anati, S. Gueron, S. P. Johnson, and V. R. Scarlata. Innovative Technology for CPU Based Attestation and Sealing. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP)*, pages 1–8, Tel-Aviv, Israel, 2013.

[4] A. Baumann, M. Peinado, and G. Hunt. Shielding applications from an untrusted cloud with haven. In *Proceedings of the 11th Symposium on Operating Systems Design and Implementation (OSDI)*, pages 267–283, Broomfield, Colorado, Oct. 2014.

[5] S. Bhatkar, D. C. DuVarney, and R. Sekar. Address obfuscation: An efficient approach to combat a broad range of memory error exploits. In *Proceedings of the 12th Usenix Security Symposium (Security)*, pages 105–120, Washington, DC, Aug. 2003.

[6] D. Bigelow, T. Hobson, R. Rudd, W. Streilein, and H. Okhravi. Timely rerandomization for mitigating memory disclosures. In *Proceedings of the 22st ACM Conference on Computer and Communications Security (CCS)*, Denver, Colorado, Oct. 2015.

[7] T. Bletsch, X. Jiang, and V. Freeh. Mitigating code-reuse attacks with control-flow locking. In *Proceedings of 27th Annual Computer Security Applications Conference (ACSAC)*, pages 353–362, 2011.

[8] C. Cadar, D. Dunbar, and D. R. Engler. Klee: Unassisted and automatic generation of high-coverage tests for complex systems programs. In *Proceedings of the 8th Symposium on Operating Systems Design and Implementation (OSDI)*, pages 209–224, San Diego, CA, Dec. 2008.

[9] P. Carbin. Intel Identity Protection Technology with PKI (Intel IPT with PKI), May 2012. White Paper, Technology Overview.

[10] M. Castro, M. Costa, and T. Harris. Securing software by enforcing data-flow integrity. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)*, pages 147–160, Seattle, WA, Nov. 2006.

[11] M. Castro, M. Costa, J.-P. Martin, M. Peinado, P. Akritidis, A. Donnelly, P. Barham, and R. Black. Fast byte-granularity software fault isolation. In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP)*, pages 45–58, Big Sky, MT, Oct. 2009.

[12] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[13] A. Chaturvedi, S. Bhatkar, and R. Sekar. Improving attack detection in host-based ids by learning properties of system call arguments. In *Proceedings of the 26th IEEE Symposium on Security and Privacy (Oakland)*, Oakland, CA, May 2005.

[14] S. Checkoway and H. Shacham. Iago Attacks: Why the System Call API is a Bad Untrusted RPC Interface. In *Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 253–264, Houston, TX, Mar. 2013.

[15] X. Chen, T. Garfinkel, E. C. Lewis, P. Subrahmanyam, C. A. Waldspurger, D. Boneh, J. Dwoskin, and

D. R. Ports. Overshadow: a virtualization-based approach to retrofitting protection in commodity operating systems. In *Proceedings of the 13th international conference on Architectural support for programming languages and operating systems*, ASPLOS XIII, pages 2–13, Seattle, WA, USA, 2008. ACM.

[16] V. Chipounov, V. Kuznetsov, and G. Candea. S2e: a platform for in-vivo multi-path analysis of software systems. In *Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Newport Beach, CA, Mar. 2011.

[17] M. Christodorescu, S. Jha, and C. Kruegel. Mining specifications of malicious behavior. In *Proceedings of the 1st India software engineering conference*, pages 5–14. ACM, 2008.

[18] B. Coppens, I. Verbauwhede, K. De Bosschere, and B. De Sutter. Practical mitigations for timing-based side-channel attacks on modern x86 processors. In *Proceedings of the 30th IEEE Symposium on Security and Privacy (Oakland)*, pages 45–60, Oakland, CA, May 2009.

[19] C. Cowan, C. Pu, D. Maier, H. Hintony, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, and Q. Zhang. StackGuard: Automatic adaptive detection and prevention of buffer-overflow attacks. In *Proceedings of the 7th Usenix Security Symposium (Security)*, San Antonio, TX, Jan. 1998.

[20] S. Crane, C. Liebchen, A. Homescu, L. Davi, P. Larsen, A.-R. Sadeghi, S. Brunthaler, and M. Franz. Readactor: Practical code randomization resilient to memory disclosure. In *Proceedings of the 36th IEEE Symposium on Security and Privacy (Oakland)*, San Jose, CA, May 2015.

[21] S. Crane, S. Volckaert, F. Schuster, C. Liebchen, P. Larsen, L. Davi, A.-R. Sadeghi, T. Holz, B. D. Sutter, and M. Franz. It's a trap: Table randomization and protection against function reuse attacks. In *Proceedings of the 22st ACM Conference on Computer and Communications Security (CCS)*, Denver, Colorado, Oct. 2015.

[22] S. Davenport and R. Ford. SGX: the good, the bad and the downright ugly, Jan. 2014. https://www.virusbtn.com/virusbulletin/archive/2014/01/vb201401-SGX.

[23] Ú. Erlingsson and F. B. Schneider. SASI enforcement of security policies: A retrospective. In *Proceedings of the New Security Paradigms Workshop*, 1999.

[24] Ú. Erlingsson, M. Abadi, M. Vrable, M. Budiu, and G. C. Necula. XFI: Software guards for system address spaces. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)*, pages 75–88, Seattle, WA, Nov. 2006.

[25] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner. Android permissions demystified. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS)*, pages 627–638, Chicago, Illinois, Oct. 2011.

[26] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 3. ACM, 2012.

[27] B. Ford and R. Cox. Vx32: Lightweight user-level sandboxing on the x86. In *Proceedings of the 2008 ATC Annual Technical Conference (ATC)*, pages 293–306, Boston, MA, June 2008.

[28] Y. Fu, Z. Lin, and K. Hamlen. Subverting systems authentication with context-aware, reactive virtual machine introspection. In *Proceedings of the 29th Annual Computer Security Applications Conference (ACSAC'13)*, New Orleans, Louisiana, December 2013.

[29] T. Garfinkel, B. Pfaff, and M. Rosenblum. Ostia: A delegating architecture for secure system call

interposition. In *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2004.

[30] C. Giuffrida, A. Kuijsten, and A. S. Tanenbaum. Enhanced operating system security through efficient and fine-grained address space randomization. In *Proceedings of the 21st Usenix Security Symposium (Security)*, pages 475–490, Bellevue, WA, Aug. 2012.

[31] J. Hiser, A. Nguyen-Tuong, M. Co, M. Hall, and J. W. Davidson. Ilr: Where'd my gadgets go? In *Proceedings of the 33rd IEEE Symposium on Security and Privacy (Oakland)*, pages 571–585, San Francisco, CA, May 2012.

[32] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo. Using innovative instructions to create trustworthy software solutions. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP)*, pages 1–8, Tel-Aviv, Israel, 2013.

[33] S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of computer security*, 6(3):151–180, 1998.

[34] Intel. Graphics Drivers Blue-ray Disc* Playback On Intel Graphics FAQ. http://www.intel.com/support/graphics/sb/CS-029871.htm, 2008. Accessed: 05/04/2015.

[35] Intel. Intel Software Guard Extensions Programming Reference (rev1), Sept. 2013. 329298-001US.

[36] Intel. Intel Software Guard Extensions Programming Reference (rev2), Oct. 2014. 329298-002US.

[37] E. Kang and D. Jackson. Designing and analyzing a flash file system with alloy. *Int. J. Software and Informatics*, 3:129–148, 2009.

[38] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall. A conundrum of permissions: installing applications on an android smartphone. In *Financial Cryptography and Data Security*, pages 68–79. Springer, 2012.

[39] S. T. King and P. M. Chen. Subvirt: Implementing malware with virtual machines. In *Proceedings of the 27th IEEE Symposium on Security and Privacy (Oakland)*, Oakland, CA, May 2006.

[40] G. Klein, K. Elphinstone, G. Heiser, J. Andronick, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood. sel4: Formal verification of an os kernel. In *Proceedings of the ACM SIGOPS 22Nd Symposium on Operating Systems Principles*, SOSP '09, pages 207–220, 2009.

[41] C. Kruegel, D. Mutz, F. Valeur, and G. Vigna. On the detection of anomalous system call arguments. In *Computer Security–ESORICS 2003*, pages 326–343. Springer, 2003.

[42] S. Kumar and E. H. Spafford. A software architecture to support misuse intrusion detection. 1995.

[43] C. Lattner and V. Adve. Llvm: A compilation framework for lifelong program analysis & transformation. In *Proceedings of the International Symposium on Code Generation and Optimization: Feedback-directed and Runtime Optimization*, CGO '04, Palo Alto, California, 2004. ISBN 0-7695-2102-9.

[44] X. Leroy. A formally verified compiler back-end. *Journal of Automated Reasoning*, 43(4):363–446, 2009.

[45] N. Li, Z. Mao, and H. Chen. Usable mandatory integrity protection for operating systems. In *Proceedings of the 28th IEEE Symposium on Security and Privacy (Oakland)*, pages 164–178, Oakland, CA, May 2007.

[46] Y. Li, J. McCune, J. Newsome, A. Perrig, B. Baker, and W. Drewry. MiniBox: A Two-Way Sandbox for x86 Native Code. In *Proceedings of the 2014 ATC Annual Technical Conference (ATC)*, pages 409–420, Philadelphia, PA, June 2014.

[47] S. McCamant and G. Morrisett. Evaluating SFI for a CISC architecture. In *Proc. USENIX Security Sym.*, 2006.

[48] J. M. McCune, B. J. Parno, A. Perrig, M. K. Reiter, and H. Isozaki. Flicker: An Execution Infrastructure for TCB Minimization. In *Proceedings of the ACM EuroSys Conference*, pages 315–328, Glasgow, Scotland, Mar. 2008.

[49] J. M. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V. Gligor, and A. Perrig. Trustvisor: Efficient tcb reduction and attestation. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, pages 143–158. IEEE Computer Society, 2010. ISBN 978-0-7695-4035-1. doi: 10.1109/SP.2010.17. URL http://dx.doi.org/10.1109/SP.2010.17.

[50] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar. Innovative instructions and software model for isolated execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP)*, pages 1–8, Tel-Aviv, Israel, 2013.

[51] G. C. Necula. Proof-carrying code. In *Proc. ACM Principles of Programming Languages*, pages 106–119, 1997.

[52] G. C. Necula, S. McPeak, S. P. Rahul, and W. Weimer. Cil: Intermediate language and tools for analysis and transformation of c programs. In *Compiler Construction*, pages 213–228. Springer, 2002.

[53] P. Ning, Y. Cui, D. S. Reeves, and D. Xu. Techniques and tools for analyzing intrusion alerts. *ACM Transactions on Information and System Security*, 7(2):274–318, 2004.

[54] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2003.

[55] V. Pappas, M. Polychronakis, and A. D. Keromytis. Smashing the gadgets: Hindering return-oriented programming using in-place code randomization. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 601–615. IEEE, 2012.

[56] B. Parno. Bootstrapping trust in a "trusted" platform. In *Proceedings of the 3rd Conference on Hot Topics in Security (HotSec)*, pages 9:1–9:6, 2008.

[57] R. Perez, R. Sailer, L. van Doorn, et al. vTPM: virtualizing the trusted platform module. In *Proceedings of the 15th Usenix Security Symposium (Security)*, pages 305–320, Vancouver, Canada, July 2006.

[58] D. E. Porter, S. Boyd-Wickizer, J. Howell, R. Olinsky, and G. C. Hunt. Rethinking the library os from the top down. In *Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 291–304, Newport Beach, CA, Mar. 2011.

[59] N. Provos. Improving host security with system call policies. In *Proc. USENIX Security Sym.*, 2003.

[60] N. Provos. Improving host security with system call policies. In *Proceedings of the 12th Usenix Security Symposium (Security)*, Washington, DC, Aug. 2003.

[61] J. Rutkowska. Thoughts on Intel's upcoming Software Guard Extensions (Part 1), Aug. 2013. http://theinvisiblethings.blogspot.com/2013/08/thoughts-on-intels-upcoming-

software.html.

[62] J. Rutkowska. Thoughts on Intel's upcoming Software Guard Extensions (Part 2), Sept. 2013. http://theinvisiblethings.blogspot.com/2013/09/thoughts-on-intels-upcoming-software.html.

[63] N. Santos, H. Raj, S. Saroiu, and A. Wolman. Using arm trustzone to build a trusted language runtime for mobile applications. In *ACM SIGARCH Computer Architecture News*, volume 42, pages 67–80. ACM, 2014.

[64] F. B. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, 2000.

[65] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich. VC3: Trustworthy Data Analytics in the Cloud using SGX. In *Proceedings of the 36th IEEE Symposium on Security and Privacy (Oakland)*, San Jose, CA, May 2015.

[66] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni. A fast automaton-based method for detecting anomalous program behaviors. In *Proceedings of the 22st IEEE Symposium on Security and Privacy (Oakland)*, pages 144–155, Oakland, CA, May 2001.

[67] A. Seshadri, M. Luk, N. Qu, and A. Perrig. Secvisor: a tiny hypervisor to provide lifetime kernel code integrity for commodity oses. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, SOSP '07, pages 335–350, Stevenson, Washington, USA, 2007. ISBN 978-1-59593-591-5.

[68] H. Shacham, M. Page, B. Pfaff, E.-J. Goh, N. Modadugu, and D. Boneh. On the effectiveness of address-space randomization. In *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)*, pages 298–307, Washington, DC, Oct. 2004.

[69] U. Steinberg and B. Kauer. NOVA: A microhypervisor-based secure virtualization architecture. In *Proceedings of the ACM EuroSys Conference*, pages 209–222, Paris, France, Apr. 2010.

[70] K. Sun, J. Wang, F. Zhang, and A. Stavrou. SecureSwitch: BIOS-assisted isolation and switch between trusted and untrusted commodity oses. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2012.

[71] L. Szekeres, M. Payer, T. Wei, and D. Song. Sok: Eternal war in memory. In *Proceedings of the 34th IEEE Symposium on Security and Privacy (Oakland)*, pages 48–62, San Francisco, CA, May 2013.

[72] P. Team. PaX address space layout randomization (ASLR), 2003.

[73] L. Van Doorn. Hardware virtualization trends. In *ACM/Usenix International Conference On Virtual Execution Environments: Proceedings of the 2 nd international conference on Virtual execution environments*, volume 14, pages 45–45, 2006.

[74] D. A. Wagner. Janus: An approach for confinement of untrusted applications. Master's thesis, U. California at Berkeley, 1999.

[75] J. Wang, A. Stavrou, and A. K. Ghosh. Hypercheck: A hardware-assisted integrity monitor. In *Recent Advances in Intrusion Detection, 13th International Symposium, RAID 2010, Ottawa, Ontario, Canada, September 15-17, 2010. Proceedings*, pages 158–177, 2010.

[76] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: alternative data models. In *Proceedings of the 20th IEEE Symposium on Security and Privacy (Oakland)*, pages 133–145, Oakland, CA, May 1999.

[77] R. Wartell, V. Mohan, K. Hamlen, and Z. Lin. Securing untrusted code via compiler-agnostic binary rewriting. In *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC'12)*, Orlando, FL, December 2012.

[78] R. Wartell, V. Mohan, K. Hamlen, and Z. Lin. Binary stirring: Self-randomizing instruction addresses of legacy x86 binary code. In *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS'12)*, Raleigh, NC, October 2012.

[79] Wikipedia. C dynamic memory allocation — wikipedia, the free encyclopedia, 2015. URL http://en.wikipedia.org/w/index.php?title=C_dynamic_memory_allocation&oldid=658580417. [Online; accessed 13-May-2015].

[80] Y. Xu, W. Cui, and M. Peinado. Controlled-channel attacks: Deterministic side channels for untrusted operating systems. In *Proceedings of the 36th IEEE Symposium on Security and Privacy (Oakland)*, San Jose, CA, May 2015.

[81] B. Yee, D. Sehr, G. Dardyk, J. B. Chen, R. Muth, T. Ormandy, S. Okasaka, N. Narula, and N. Fullagar. Native Client: A sandbox for portable, untrusted x86 native code. In *Proc. IEEE Sym. Security and Privacy*, pages 79–93, 2009.

[82] J. Zeng, Y. Fu, and Z. Lin. Pemu: A pin highly compatible out-of-vm dynamic binary instrumentation framework. In *Proceedings of the 11th Annual International Conference on Virtual Execution Environments*, Istanbul, Turkey, March 2015.

[83] C. Zhang, T. Wei, Z. Chen, L. Duan, L. Szekeres, S. McCamant, D. Song, and W. Zou. Practical control flow integrity and randomization for binary executables. In *Proceedings of the 34th IEEE Symposium on Security and Privacy (Oakland)*, pages 559–573, San Francisco, CA, May 2013.

[84] F. Zhang, J. Chen, H. Chen, and B. Zang. Cloudvisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 203–216, Cascais, Portugal, 2011. ACM. ISBN 978-1-4503-0977-6.

[85] M. Zhang and R. Sekar. Control flow integrity for COTS binaries. In *Proceedings of the 22th Usenix Security Symposium (Security)*, pages 337–352, Washington, DC, Aug. 2013.

Biographical Sketch: Taesoo Kim

■ Professional Preparation

Massachusetts Institute of Technology	EECS	Ph.D.	2014	Cambridge, MA
Massachusetts Institute of Technology	EECS	S.M.	2011	Cambridge, MA
Korea Advanced Institute of Science and Technology	CS	B.S.	2009	Daejeon, South Korea
Korea Advanced Institute of Science and Technology	EE	B.S.	2009	Daejeon, South Korea

■ Appointments

08/2014–present	Assistant Professor, School of Computer Science, Georgia Institute of Technology
06/2014–08/2014	Visiting Scholar, Computer Science and Engineering, University of Washington
07/2012–08/2012	Research Intern, Samsung Electronics
01/2012–05/2012	Co-founder & Programmer, Nerati (now Compass)
06/2010–09/2014	Research Intern, Microsoft Research

■ Teaching

02/2016–08/2016	Design Operating Systems (scheduled, CS 3210)
08/2015–12/2015	Information Security Lab (scheduled, CS 6265)
08/2014–12/2014	Topics in Building Secure Systems (8803-BSS, eval: 4.8/5.0, link)
09/2012–12/2012	Teaching Assistant: Computer Systems Security (MIT 6.858, eval: 6.3/7.0, link)

■ Products

Five Products Most Closely Related to the Proposed Project:

1. Kangjie Lu, Chengyu Song, Byoungyoung Lee, Simon P. Chung, Taesoo Kim, and Wenke Lee. ASLR-Guard: Stopping Address Space Leakage for Code Reuse Attacks (to appear). In *Proceedings of The 22nd ACM Conference on Computer and Communications Security (CCS 2015),* Denver, CO, October 2015.

2. Byoungyoung Lee, Chengyu Song, Taesoo Kim, and Wenke Lee. Type Casting Verification: Stopping an Emerging Attack Vector. In *Proceedings of the 24th USENIX Security Symposium (Security 2015),* Washington, DC, August 2015.

3. Byoungyoung Lee, Chengyu Song, Yeongjin Jang, Tielei Wang, Taesoo Kim, Long Lu, and Wenke Lee. Preventing Use-after-free with Dangling Pointers Nullification. In *Proceedings of the 2015 Network and Distributed System Security Symposium (NDSS 2015),* San Diego, CA, February 2015.

4. Byoungyoung Lee, Long Lu, Tielei Wang, Taesoo Kim, and Wenke Lee. From Zygote to Morula: Fortifying Weakened ASLR on Android. In *Proceedings of the 35th IEEE Symposium on Security and Privacy (Oakland 2014),* San Jose, CA, May 2014.

5. Taesoo Kim, Marcus Peinado, and Gloria Mainar-Ruiz. System-Level Protection Against Cache-based Side Channel Attacks in the Cloud. In *Proceedings of the 21st USENIX Security Symposium (Security 2012),* Bellevue, WA, August 2012.

Five Other Significant Products:

1. Changwoo Min, Sanidhya Kashyap, Byoungyoung Lee, Chengyu Song, and Taesoo Kim. Cross-checking Semantic Correctness: The Case of Finding File System Bugs. In *Proceedings of the 25th ACM Symposium on Operating Systems Principles (SOSP 2015),* Monterey, CA, October 2015 (to appear).

2. Haogang Chen, Taesoo Kim, Xi Wang, M. Frans Kaashoek, and Nickolai Zeldovich. Identifying Information Disclosure in Web Applications with Retroactive Auditing. In *Proceedings of the 11th Symposium on Operating Systems Design and Implementation (OSDI 2014)*, Broomfield, CO, October 2014.

3. Ramesh Chandra, Taesoo Kim, and Nickolai Zeldovich. Asynchronous Intrusion Recovery for Interconnected Web Services. In *Proceedings of the 24th ACM Symposium on Operating Systems Principles (SOSP 2013),* Farmington, PA, November 2013.

4. Taesoo Kim, Ramesh Chandra, and Nickolai Zeldovich. Efficient Patch-based Auditing for Web Application Vulnerabilities. In *Proceedings of the 10th Symposium on Operating Systems Design and Implementation (OSDI 2012)*, Hollywood, CA, October 2012.

5. Taesoo Kim, Xi Wang, Nickolai Zeldovich, and M. Frans Kaashoek. Intrusion Recovery using Selective Re-execution. In *Proceedings of the 9th Symposium on Operating Systems Design and Implementation (OSDI 2010)*, Vancouver, Canada, October 2010.

■ Synergistic Activities

1. **Advising and research.** I founded and lead the *Systems Software & Security Lab*. We have made all of our projects publicly available, and actively contributed to open source communities, including CRIU, Firefox, LLVM, Android and Linux. For example, our recent bug-finding tools found 140 previously unknown bugs in Linux file-systems, and several in Firefox and GNU Libc. As a result, we have been awarded several bug bounties from Mozilla and other companies, and awarded 2015 Internet Defense Prize ($100k prize) from Facebook and USENIX.
2. **Industrial impacts.** Dr. Kim's thesis work become a basis of a startup company that he and three other colleagues at MIT and Stanford started with $2 millions initial investment from Bain Capital Ventures in 2011.
3. **Program committee.** SYSTOR 2016, INFOCOM 2016, CCS 2015, Usenix Security 2015, APSys 2015, WISA 2013.
4. **Journal reviewer.** ACM Transactions on information and System Security (TOISS 2014/2015), IEEE/ACM Transactions on Networking (ToN 2013), Security and Communication Networks (SCN 2014)
5. **Web admin.** Eurosys 2012

■ Collaborators & Other Affiliations

1. **Collaborators (26 total):**Thomas Anderson (University of Washington), Alexandra Boldyreva (Georgia Tech), Soham Desai (Intel), Young Ik Eom (Sungkyunkwan University), Hadi Esmaeilzadeh (Georgia Tech), Dongsu Han (KAIST), Bill Harris (Georgia Tech), Prerit Jain (Oracle), M. Frans Kaashoek (MIT), Brent Kang (KAIST), Yongdae Kim (KAIST), Arvind Krishnamurthy (University of Washington), Sang-Won Lee (Sungkyunkwan University), Wenke Lee (Georgia Tech), Long Lu (SUNY), Gloria Mainar-Ruiz (Microsoft Research), Robert Morris (MIT), Todd C Mowry (CMU), Onur Mutlu (CMU), Mayur Naik (Georgia Tech), Marcus Peinado (Microsoft Research), Tielei Wang (Georgia Tech), Xi Wang (University of Washington), David Wetherall (Google), Xinyu Xing (Pennsylvania State University), Nickolai Zeldovich (MIT).
2. **Graduate Advisors (2 total):** Nickolai Zeldovich (MIT), M. Frans Kaashoek (MIT)
3. **Thesis Advisor and Postgraduate-Scholar Sponsor (10 total):** Changwoo Min (Georgia Tech, Postgraduate-Scholar), Sanidhya Kashyap (Georgia Tech, PhD), Yang Ji (Georgia Tech, PhD), Meng Xu (Georgia Tech, PhD), Steffen Maass (Georgia Tech, PhD), Insu Yun (Georgia Tech, PhD), YeongJin Jang (Georgia Tech, PhD, co-advised), Chengyu Song (Georgia Tech, PhD, co-advised), Kangjie Lu (Georgia Tech, PhD, co-advised), and Byoungyoung Lee (Georgia Tech, PhD, co-advised).

Updated: 15th September, 2015

Zhiqiang Lin
Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080
zhiqiang.lin@utdallas.edu, http://www.utdallas.edu/∼zhiqiang.lin, (972) 883–4244

(a) Professional Preparation

Ph.D. in Computer Science, Purdue University, USA — August 2011
M.S. in Computer Science, Nanjing University, China — December 2006
B.S. in Computer Science, Nanjing University of Posts & Telecommunications, China — July 2002

(b) Appointments

Assistant Professor, Department of Computer Science, University of Texas at Dallas — Sept 2011 -
Research Assistant, Department of Computer Science, Purdue University — Jan 2007 - Aug 2011
Summer Intern, CSL, SRI International Menlo Park CA — May 2008 - Aug 2008

(c) Selected Publications

Products: Five Most Relevant Products and Publications Related to the Proposal

USENIX-SEC — F. Peng, Z. Deng, X. Zhang, D. Xu, **Z. Lin**, and Z. Su. "X-Force: Force-Executing Binary Programs for Security Applications", In *Proceedings of the USENIX Security Symposium*, August 2014.

NDSS — A. Saberi, Y. Fu, and **Z. Lin**, "Hybrid-Bridge: Efficiently Bridging the Semantic-Gap in Virtual Machine Introspection via Decoupled Execution and Training Memoization", In *Proceedings of the Network and Distributed System Security Symposium*, February 2014.

CCS — J. Zeng, Y. Fu, K. Miller, **Z. Lin**, X. Zhang, and D. Xu, "Obfuscation-resilient Binary Code Reuse through Trace-oriented Programming", in *Proceedings of the 20th ACM Conference on Computer and Communications Security*, Berlin, Germany, November 2013.

CCS — R. Wartell, V. Mohan, K. Hamlen, and **Z. Lin**, "Binary Stirring: Self-randomizing Instruction Addresses of Legacy x86 Code", In *Proceedings of the 19th ACM Conference on Computer and Communications Security*, October 2012

Oakland — Y. Fu and **Z. Lin**, "Space Traveling across VM: Automatically Bridging the Semantic Gap in Virtual Machine Introspection via Online Kernel Data Redirection", In *Proceedings of the 33rd IEEE Symposium on Security and Privacy*, May 2012

Five Other Significant Products and Publications

NDSS — **Z. Lin**, J. Rhee, C. Wu, X. Zhang, and D. Xu, "DIMSUM: Discovering Semantic Data of Interest from Un-mappable Memory with Confidence", In *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, February 2012.

NDSS — **Z. Lin**, J. Rhee, X. Zhang, D. Xu, and X. Jiang, "SigGraph: Brute Force Scanning of Kernel Data Structure Instances Using Graph-based Signatures", In *Proceedings of the 18th Annual Network & Distributed System Security Symposium*, February 2011.

NDSS — **Z. Lin**, X. Zhang, D. Xu, "Automatic Reverse Engineering of Data Structures from Binary Execution", In *Proceedings of the 17th Proceedings of Network & Distributed Systems Security Symposium*, February 2010.

FSE **Z. Lin**, and X. Zhang, "Deriving Input Syntactic Structure From Execution", In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, November 2008. (An extended journal version is published at *IEEE Transactions on Software Engineering*, 36(5), 2010).

NDSS **Z. Lin**, X. Jiang, D. Xu, and X. Zhang. "Automatic protocol format reverse engineering through context-aware monitored execution", In *Proceedings of 15th Annual Network & Distributed System Security Symposium*, February 2008.

(d) Synergistic Activities

Panelist	NSF proposal review panels	2012
Program Co-chair	IPCCC	2013
	NGMAD (In conjunction with ACSAC)	2013
TPC Member	CCS, NDSS, ASIACCS, DFRWS, PST, SECURECOMM	2015
	CCS, ICDCS, ACSAC, CCGrid, ASIACCS, DFRWS	2014
	CCGrid, ASIACCS, DFRWS, PST, IPCCC, INSCRYPT	2013
	IPCCC, ARES, SPE, HASP	2012
Journal Reviewer	ACM TISSEC, ACM TACO, IEEE TC, IEEE TDSC, IEEE TIFS, IEEE TKDE, ACM Computing Survey, IEEE Computer Architecture Letters, Computer Networks Journal, Computer & Security, Journal of Systems and Software	
Web Administrator	SECURECOMM	2015

(e) Collaborators

Collaborators (18)	David Brumley (CMU), Juan Caballero (IMDEA), Haibo Chen (SJTU), Guofei Gu (Texas A&M), Kevin Hamlen (UT Dallas), Xuxian Jiang (Qihoo), Murat Kantarcioglu (UT Dallas), Ashish Kundu (IBM), Latifur Lkhan (UT Dallas), Charles McFarland (Intel), Junghwan Rhee (NEC Research Lab), Ryan Riley (Qatar University), Kevin Roundy (Symantec), Weidong Shi (University of Houston), Bhavani Thuraisingham (UT Dallas), Shouhuai Xu (UT San Antonio), Heng Yin (Syracuse), Mingwei Zhang (Intel)
Graduate Advisors (2)	Dongyan Xu (Purdue), and Xiangyu Zhang (Purdue)
Thesis Advisor (9)	Erick Bauman, Swarup Chandra, Yangchun Fu, Yufei Gu, Raul Quinonez, Huibo Wang, Wubing Wang, Junyuan Zeng (FireEye), Chaoshun Zuo
Academic Visitor (1)	Alireza Saberi (Microsoft)

SUMMARY PROPOSAL BUDGET

YEAR 1

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Georgia Tech Research Corporation

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Taesoo Kim

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Taesoo Kim - Assistant Professor**	0.25	0.00	0.00	**2,833**	
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.25	0.00	0.00	**2,833**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**2**) GRADUATE STUDENTS				**58,537**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**61,370**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**2,606**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**63,976**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**2,400**	
2. FOREIGN				**3,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**2,081**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**5,205**	
5. SUBAWARDS				**0**	
6. OTHER				**38,595**	
TOTAL OTHER DIRECT COSTS				**45,881**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**115,257**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **Facilities and Administration (Rate: 55.9000, Base: 76662)**					
TOTAL INDIRECT COSTS (F&A)				**42,854**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**158,111**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**158,111**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Taesoo Kim

ORG. REP. NAME*
Tanya Blackwell

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

YEAR 2

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Georgia Tech Research Corporation

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Taesoo Kim

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Taesoo Kim - Assistant Professor**	0.25	0.00	0.00	**2,917**	
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.25	0.00	0.00	**2,917**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**2**) GRADUATE STUDENTS				**60,293**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**63,210**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**2,684**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**65,894**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**2,400**	
2. FOREIGN				**3,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**2,081**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**5,361**	
5. SUBAWARDS				**0**	
6. OTHER				**41,682**	
TOTAL OTHER DIRECT COSTS				**49,124**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**120,418**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **Facilities and Administration (Rate: 55.9000, Base: 78736)**					
TOTAL INDIRECT COSTS (F&A)				**44,013**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**164,431**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**164,431**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Taesoo Kim

ORG. REP. NAME*
Tanya Blackwell

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

YEAR 3

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Georgia Tech Research Corporation

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Taesoo Kim

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Taesoo Kim - Assistant Professor**	0.25	0.00	0.00	**3,005**	
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.25	0.00	0.00	**3,005**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**2**) GRADUATE STUDENTS				**62,102**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**65,107**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**2,765**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**67,872**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**2,400**	
2. FOREIGN				**3,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**2,081**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**5,522**	
5. SUBAWARDS				**0**	
6. OTHER				**45,017**	
TOTAL OTHER DIRECT COSTS				**52,620**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**125,892**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **Facilities and Administration (Rate: 55.9000, Base: 80875)**					
TOTAL INDIRECT COSTS (F&A)				**45,209**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**171,101**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**171,101**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Taesoo Kim

ORG. REP. NAME*
Tanya Blackwell

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

YEAR 4

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Georgia Tech Research Corporation

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Taesoo Kim

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Taesoo Kim - Assistant Professor**	0.25	0.00	0.00	**3,095**	
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.25	0.00	0.00	**3,095**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**2**) GRADUATE STUDENTS				**63,965**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**67,060**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**2,848**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**69,908**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**2,400**	
2. FOREIGN				**3,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**2,081**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**5,687**	
5. SUBAWARDS				**0**	
6. OTHER				**48,618**	
TOTAL OTHER DIRECT COSTS				**56,386**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**131,694**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **Facilities and Administration (Rate: 55.9000, Base: 83076)**					
TOTAL INDIRECT COSTS (F&A)				**46,439**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**178,133**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**178,133**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Taesoo Kim

ORG. REP. NAME*
Tanya Blackwell

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

Cumulative

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Georgia Tech Research Corporation

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Taesoo Kim

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Taesoo Kim - Assistant Professor**	1.00	0.00	0.00	**11,850**	
2.					
3.					
4.					
5.					
6. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	1.00	0.00	0.00	**11,850**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**8**) GRADUATE STUDENTS				**244,897**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**256,747**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**10,903**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**267,650**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**9,600**	
2. FOREIGN				**12,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**8,324**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**21,775**	
5. SUBAWARDS				**0**	
6. OTHER				**173,912**	
TOTAL OTHER DIRECT COSTS				**204,011**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**493,261**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)					
TOTAL INDIRECT COSTS (F&A)				**178,515**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**671,776**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**671,776**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Taesoo Kim

ORG. REP. NAME*
Tanya Blackwell

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

C ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

Georgia Tech Research Corporation
School of Computer Science/Georgia Institute of Technology
Budget Justification

Senior Personnel:
Dr. Kim is requesting for .25 month salary support per year for the duration of the project. There is a 3% salary increase budgeted for each year starting in July.

Other Personnel:
Funds are budgeted to support two Graduate Research Assistants (GRAs) at the 50% PhD rate. There is a 3% salary increase budgeted for each year starting in July.

Fringe Benefits:
A fringe benefit rate of 30% has been applied to the PI's salary. No fringe rate is required for graduate student salaries. A cost of 3% of GRA salaries is added for student health care costs.

M&S:
Funds are budgeted for materials and supplies related to the project. The materials include research related software, books, printing, publications, small non-inventoried equipment items such as mobile devices and peripherals, lab supplies, cloud hosting, and conference registrations. The small non-inventoried equipment items are necessary to complete the scope of the work of this project and will be dedicated to the benefit of this project over the project performance period.

Travel:
The travel budget includes funding for the PI and the GRAs to attend directly related conferences, workshops, and meetings where the results of the proposed research will be presented and/or discussed.

Other Direct Cost: Other:
The budget also includes the following.

- Computer Services provided by the College of Computing have the costs computed at the ONR approved rate of $412.50 per full time employee each month; these funds cover the costs associated with network fees and provision of computer services by the College of Computing for the PI, Co-PI, and post doc supported on the grant.

Indirect Costs:
Indirect costs of 55.9% are charged for all direct costs except graduate student tuition. The rates for the cost estimates could be found at http://www.osp.gatech.edu/rates/.

SUMMARY PROPOSAL BUDGET

YEAR 1

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
University of Texas at Dallas

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Zhiqiang Lin

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Zhiqiang Lin - PI**	0.00	0.00	0.05	**569**	
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	0.05	**569**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**3**) GRADUATE STUDENTS				**70,200**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**70,769**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**10,644**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**81,413**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**6,000**	
2. FOREIGN				**3,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**6,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**6,000**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**96,413**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **MTDC (Rate: 53.0000, Base: 96413)**					
TOTAL INDIRECT COSTS (F&A)				**51,099**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**147,512**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**147,512**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Zhiqiang Lin

ORG. REP. NAME*
Emily lacy

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

YEAR 2

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
University of Texas at Dallas

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Zhiqiang Lin

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Zhiqiang Lin - PI**	0.00	0.00	0.05	**586**	
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	0.05	**586**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**3**) GRADUATE STUDENTS				**72,306**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**72,892**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**10,963**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**83,855**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**6,000**	
2. FOREIGN				**3,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**0**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**0**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**92,855**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **MTDC (Rate: 53.0000, Base: 92855)**					
TOTAL INDIRECT COSTS (F&A)				**49,213**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**142,068**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**142,068**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Zhiqiang Lin

ORG. REP. NAME*
Emily lacy

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

YEAR 3

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
University of Texas at Dallas

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Zhiqiang Lin

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Zhiqiang Lin - PI**	0.00	0.00	0.75	**9,059**	
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	0.75	**9,059**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**2**) GRADUATE STUDENTS				**49,650**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**58,709**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**9,259**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**67,968**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**6,000**	
2. FOREIGN				**3,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**0**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**0**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**76,968**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **MTDC (Rate: 53.0000, Base: 76968)**					
TOTAL INDIRECT COSTS (F&A)				**40,793**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**117,761**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**117,761**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Zhiqiang Lin

ORG. REP. NAME*
Emily lacy

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

YEAR 4

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
University of Texas at Dallas

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Zhiqiang Lin

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Zhiqiang Lin - PI**	0.00	0.00	0.75	**9,331**	
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	0.75	**9,331**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**2**) GRADUATE STUDENTS				**51,140**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**60,471**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**9,537**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**70,008**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**6,000**	
2. FOREIGN				**3,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**0**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**0**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**79,008**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **MTDC (Rate: 53.0000, Base: 79007)**					
TOTAL INDIRECT COSTS (F&A)				**41,874**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**120,882**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**120,882**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Zhiqiang Lin

ORG. REP. NAME*
Emily lacy

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

Cumulative

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION: **University of Texas at Dallas**

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR: **Zhiqiang Lin**

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Zhiqiang Lin - PI**	0.00	0.00	1.60	**19,545**	
2.					
3.					
4.					
5.					
6. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	1.60	**19,545**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**10**) GRADUATE STUDENTS				**243,296**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**262,841**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**40,403**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**303,244**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. U.S. POSSESSIONS)				**24,000**	
2. FOREIGN				**12,000**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
TOTAL NUMBER OF PARTICIPANTS (**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**6,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**6,000**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**345,244**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)					
TOTAL INDIRECT COSTS (F&A)				**182,979**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**528,223**	
K. SMALL BUSINESS FEE				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				**528,223**	

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI/PD NAME: **Zhiqiang Lin**

ORG. REP. NAME*: **Emily lacy**

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

C ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

Budget Justification from UT Dallas

Salaries and Wages. Salaries for all personnel are based upon current University of Texas at Dallas (UTD) academic and staff salary scales. Salaries for the Principal Investigator (PI) budget calculations include a 3% annual increase.

Zhiqiang Lin, the PI from UTD, will make a strong commitment to the proposed work. He will be responsible for overall project direction and coordination, for assuring successful project completion, including submission of progress reports, supervision of the graduate students, dissemination of research results, outreach and educational activities. We request summer salary support at a level of 0.05 month in year one and year two, and 0.75 month in year three and year four for this project.

We request salary support for three graduate research assistants (RA) in year one and year two, and two RAs in year three and year four, respectively, at 50% time during the academic year and at 100% time during the summer months. The role of the graduate research assistants includes engaging in the proposed research tasks and system development efforts, participating regular meetings and the research publications, and presenting the research results in academic conferences.

Fringe Benefits. Employee benefits were estimated in accordance to commonly applied or historical rates used. UTD fringe benefit rates are 20% of salary and 15% for graduate students during the academic year and summer months.

Travel. A total of two domestic and one international trip are requested for the involved graduate students or the PIs to attend the technical conferences (e.g., Oakland, CCS, USENIX Security, NDSS, SOSP, OSDI, etc.) and workshops that are relevant to this proposed research and for scientific exchange. Domestic trips are estimated at $2,000 each round-trip and foreign trips are at $3,000 each round-trip. There will be also one domestic trip to visit the leading institute Georgia Tech, for student exchange or other research meetings, and one domestic trip for SaTC PI Meeting. Expenses include estimates for airfare, ground transportation, hotel accommodations, registration for conferences and workshops (if applicable), and per diem. More specific itemized budget is presented in Table 1.

Destination	Registration	Airfare	Ground Transportation	Hotel Accommodations	Meals (Per Diem)	Total
Domestic Conference	700	600	100	500	200	2100
Domestic Conference	700	400	100	500	200	1900
Leading Institute Site Visit	-	400	100	400	200	1100
SaTC PI Meeting	-	500	50	250	100	900
International Conference (Europe or Asia)	700	1300	200	500	300	3000

Table 1: Detailed Budget for the Conference Travel.

Materials and Supplies. We request $6,000 for purchasing two desktops and one server machine with the new Intel SGX chips. This will be used by the involved graduate research assistants to develop and evaluate the research prototypes for this project.

Indirect Costs. Indirect costs were estimated in accordance with UTDs rate agreement, which was approved by DHHS, the Federal Cognizant Audit Agency for UTD on 7/24/12. The organized research F&A cost rate of 53% MTDC was used based on the nature of the proposed work.

Taesoo Kim
Assistant Professor
Georgia Institute of Technology
Current & Pending Support

CURRENT:

Title: Study on Manycore Scalability of the Next-Generation Operating Systems
Amount: $330,000
Location: Georgia Institute of Technology
Sponsor: Electronics and Telecommunications Research Institute
Award Period: 03/01/2015 – 02/28/2016
Person-Person Months: Year 1: 2 months

Title: SaTC-EDU: EAGER: Big Data and Security: Educating The Next-Generation Security Analysts
Amount: $300,000
Location: Georgia Institute of Technology
Sponsor: National Science Foundation
Award Period: 01/01/2015 - 12/31/2016
Person-Person Months: Years 1-2: 1.0 month

Title: BFT++: Attack Tolerance in Hard Real-Time Systems
Amount: $1,245,720
Location: Georgia Institute of Technology
Sponsor: Office of Naval Research
Award Period: 01/01/2015 - 12/31/2017
Person-Person Months: Years 1-3: 1.0 month

Title: THEIA: Tagging and Tracking of Multi-level Host Events for Transparent Computing and Information Assurance
Amount: $4,253,126
Location: Georgia Institute of Technology
Sponsor: DARPA
Award Period: 07/01/2015-06/30/2018
Person-Person Months: 2 months

Title: Concolic Testing to Improve Software Security
Amount: $89,000
Location: Georgia Institute of Technology
Sponsor: National Security Research Institute
Award Period: 07/01/2015 – 06/30/2016
Person-Person Months: Years 1: 0.5 month

Title: Educating the Next Generation Computer Scientists: Curriculum Development for Secure Big Data Processing
Amount: $7,000
Location: Georgia Institute of Technology
Sponsor: Georgia Institute of Technology
Award Period: 07/01/2015 – 06/30/2016
Person-Person Months: Year 1: 0 month

PENDING:

Title: TWC: Medium: Collaborative: Systems, Tools, and Techniques for Executing, Managing, and Securing SGX Programs
Amount: $671,776
Location: Georgia Institute of Technology
Sponsor: NSF
Award Period: 06/01/2016 – 05/31/2020
Person-Person Months: Year 1-4: 0.25 month

Title: CAREER: Rebootless Operating System Update and Validation via Application Checkpoint-and-Restart
Amount: $493,805
Location: Georgia Institute of Technology
Sponsor: NSF
Award Period: 01/01/2016 – 12/31/2020
Person-Person Months: Year 1-5: 0.75 month

Current and Pending Support

(See GPG Section II.C.2.h for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Investigator: Zhiqiang Lin	Other agencies (including NSF) to which this proposal has been/will be submitted.

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: Towards Fundamental and Binary-Centric Techniques for Kernel Malware Defense

Source of Support: AFOSR
Total Award Amount: $ 360,157 Total Award Period Covered: 09/01/14 - 08/31/17
Location of Project: University of Texas at Dallas
Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 1.00

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: UTD SFS Renewal: Growing a Cybersecurity Community thru SFS Scholarship Program at UTD (PI Kamil Sarac, Co-PI Alvaro Cardenas, Kevin Hamlen, Zhiqiang Lin, Bhavani)

Source of Support: NSF
Total Award Amount: $ 3,931,358 Total Award Period Covered: 09/01/14 - 08/31/19
Location of Project: University of Texas at Dallas
Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 0.25

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: Automated, Binary Evidence-Based Attribution of Software Attacks (PI Kevin Hamlen, Co-PI Latifur Khan and Zhiqiang Lin)

Source of Support: AFOSR
Total Award Amount: $ 613,443 Total Award Period Covered: 07/01/14 - 06/30/17
Location of Project: University of Texas at Dallas
Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 1.00

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: CAREER: A Dual-VM Binary Code Reuse Based Framework for Automated Virtual Machine Introspection

Source of Support: NSF
Total Award Amount: $ 535,054 Total Award Period Covered: 09/01/15 - 08/31/20
Location of Project: University of Texas at Dallas
Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 1.00

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: Deception-enabled Interactive Software for Active Cyber Defense (PI Kevin Hamlen, Co-PI Latifur Khan and Zhiqiang Lin)

Source of Support: NSA
Total Award Amount: $ 286,592 Total Award Period Covered: 09/01/15 - 08/31/16
Location of Project: University of Texas at Dallas
Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Summ: 1.00

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

Current and Pending Support

(See GPG Section II.C.2.h for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Investigator: Zhiqiang Lin	Other agencies (including NSF) to which this proposal has been/will be submitted.

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: Virtual Laboratory and Curriculum Development for Secure Mobile Computing (PI Latifur Khan, Co-PI Bhavani Thuraisingham and Zhiqiang Lin)

Source of Support: NSF

Total Award Amount: $ 299,634 Total Award Period Covered: 09/15/15 - 08/31/17

Location of Project: University of Texas at Dallas

Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 0.70

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: TWC: Medium: Collaborative: Systems, Tools, and Techniques for Executing, Managing, and Securing SGX Programs (This Proposal)

Source of Support: NSF

Total Award Amount: $ 528,223 Total Award Period Covered: 06/01/16 - 05/31/20

Location of Project: University of Texas at Dallas

Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 0.05

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Summ:

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

Georgia Tech and the College of Computing

Facilities, Equipment, and Other Resources

Georgia Tech

Located in Atlanta, Georgia, the Georgia Institute of Technology is one of the top research universities in the United States. Georgia Tech is a science and technology-focused learning institute renowned for our deeply held commitment to improving the human condition. Our faculty and students are solving some of the world's most pressing challenges: clean and sustainable energy; disease diagnosis and treatment; and national defense and security, among others. Georgia Tech is an innovative intellectual environment with nearly 1,000 full-time instructional faculty and more than 21,500 undergraduate and graduate students.

Our bachelor's, master's and doctoral degree programs are consistently recognized among the best. Georgia Tech students are equipped for success in a world where technology touches every aspect of our daily lives. Degrees are offered through the Institute's six colleges: Architecture, Computing, Engineering, Sciences, Scheller College of Business, and the Ivan Allen College of Liberal Arts. Year after year, Georgia Tech is consistently the only technological university ranked in *U.S. News & World Report's* listing of America's top ten public universities. In addition, our College of Engineering is consistently ranked in the nation's top five by *U.S. News*. In terms of producing African American engineering graduates, *Diverse: Issues in Higher Education* ranks Tech No. 1 at the doctoral level and No. 2 at the bachelor's level, based on the most recent rankings for 2012. These impressive national rankings reflect the academic prestige long associated with the Georgia Tech curriculum.

The College of Computing at Georgia Tech:

The College of Computing at Georgia Tech is a national leader in the research and creation of real-world computing breakthroughs that drive social and scientific progress. With its graduate CS program ranked 9th (2014) nationally by *U.S. News and World Report*, the College's unconventional approach to education is pioneering the new era of computing by expanding the horizons of traditional computer science students through interdisciplinary collaboration and a focus on human centered solutions.

The College resides and operates computing facilities in three buildings (College of Computing Building, Technology Square Research Building, and Klaus Advanced Computing Building), including over 3,500 computers in over 50 networks servicing 3 Schools, 8 Research Centers, and 60 Research Labs. Our data centers hosts more than 900 servers of various makes (Dell, HP, IBM, Penguin Computing, and SuperMicro), most of which are multi-processor, multi-core machines, providing over 1 PB of networked disk storage. There are 16 Linux-based high performance computing clusters totaling more than 1,000 physical servers and 6,000 computing processors/cores. All of the College's facilities are linked via local area networks that provide 1 Gigabit per second (Gbps) to the desktop. The College's network employs an internal high-performance, 10 Gbps Ethernet backbone to each of its buildings with external connectivity to the campus network by a 10 Gbps Ethernet uplink. The Georgia Tech network (GTNet) is an Ethernet based IP network spanning the 150 buildings on the main campus in Atlanta, as well as remote campuses in Savannah, GA, and Metz, France. Internet services are purchased from transit providers as well as connections to research networks and transit peering services. Georgia Tech has peering with Peachnet, Southern Crossroads (SoX), TransitRail, Cogent, and Qwest. Depending on which connection is used, traffic may be going over a 10 gigabit, 1 gigabit, or 100 megabit link. The Georgia Tech Research Network, through its peering with SoX, has connectivity to NLR Packetnet, Internet2 network, Oak Ridge National Labs, the Department of Energy's Energy Sciences Network (ESNet), NCREN, MREN, FLR, LONI, and 3ROX, as well as other SoX participants in the SouthEast.

Georgia Tech and the College of Computing

Facilities, Equipment, and Other Resources

Buildings

The **College of Computing Building (CCB)** houses administrative offices for the College, instructional classrooms and labs, and the Institute for Robotics & Intelligent Machines (IRIM) at Georgia Tech, as well as meeting space for undergraduate and graduate student organizations. CCB is the instructional center of the College, housing 9 classrooms and 3 instructional computer labs with over 30 seats to service CS courses that require special software or capabilities not readily available in the general clusters provided by OIT. The instructional labs are available to students taking a CS course requiring specialized resources. A spacious Commons Area provides ample seating and computer networking which fosters both formal and informal learning opportunities and collaboration. A highly visible conference room is equipped with a Cisco TelePresence C40 system and is available to all faculty, staff and students for conducting meetings with remote collaborators. CCB also houses a 2000 sq. ft. data center providing over 500 Kilowatts of power and cooling capacity for the College's research and instructional computational servers. The building's advanced infrastructure provides 1 Gbps networking to all ports with a 10 Gbps uplink to the campus network as well as high-density 802.11n wireless networking support.

The **Technology Square Research Building (TSRB)** is located in the innovative and pedestrian-friendly, mixed-use Technology Square district of Georgia Tech and is home to the College's School of Interactive Computing, the GVU Center as well as over 15 CoC research labs spanning multiple research groups including Human Computer Interaction, Cognitive Science, Mobile Robotics, Graphics and Animation, Information Visualization, Learning Sciences and Technology, Computing Education, Social Computing, Ubiquitous and Wearable Computing, and Virtual and Augmented Environments. TSRB also houses state of the art conference facilities that accommodate several of the College's special events, lectures and meetings. The College manages a 400 sq. ft. data center in the building providing 100 Kilowatts of power and cooling capacity for several research computational servers. The building's advanced infrastructure provides 1 Gbps networking to all ports with a 10 Gbps uplink to the campus network as well as high-density 802.11n wireless networking support.

The **Klaus Advanced Computing Building (KACB)**, dedicated in 2006, is located in the heart of the Georgia Tech campus and houses some of the most advanced computing labs and innovative educational technology in the world. The 414,000 square-foot building consists of some 70 research laboratories, 6 instructional labs, 5 large classrooms and a 200-seat auditorium. The building has a substantial number of environmental and

sustainable features achieving the prestigious LEED Gold rating from the U.S. Green Building Council. Environmentally friendly features include creative use of the 6-acre urban campus site to preserve over 50 percent of the site as green space, a storm water collection system to provide water for irrigation, energy efficient heating, cooling and lighting systems, and extensive use of recyclable materials.

KACB is home to the College's School of Computer Science, the School of Computational Science and Engineering, 6 research centers (GTISC, CERCS, C21U, Fodva, IDH, and ARC), over 20 CoC research labs spanning multiple research groups including High Performance Computing, Information Security, Software Engineering, Databases, Systems, Theory, Computer Architecture, Networking, Programming/Algorithms, Data Analytics, and Embedded Systems. KACB houses state of the art conference facilities that accommodate several of the College's special events, lectures and meetings. The building features open collaboration spaces, study lounges, conference rooms and graduate student offices, all with ample power and networking ports. All conference rooms are equipped with projection technology, table networking and power. A highly visible conference room is equipped with a Polycom HDX 8000 video conferencing system and is available to all faculty, staff and students for conducting meetings with remote collaborators. The College manages a 500 sq. ft. data center in the building providing 80 Kilowatts of power and cooling capacity for critical enterprise servers. The building's advanced infrastructure provides 1 Gpbs networking to all ports with a 10 Gbps uplink to the campus network as well as high-density 802.11n wireless networking support.

Instructional Facilities

In addition to general instructional facilities provided by the Institute, the College of Computing provides specialized instructional facilities for it's advanced curriculum needs. All of the College's instructional labs and servers are located in the College of Computing Building.

- **Shuttles UNIX Remote Access:** Supporting general-purpose UNIX shell remote access, a 5-node SuperMicro Server cluster (2 x 2.4 GHz Intel Xeon E5620 4-core, 24 GB RAM, 80 GB SSD disk).
- **Digital Media and Gaming Lab:** Supporting graphics, digital media and gaming courses, a 14-seat cluster of 12 SuperMicro workstations (4-core, 2.4GHz Intel Xeon X3430, 16GB RAM) workstations running Windows and 2 Apple 27" iMac workstations (4-core, 3.4GHz Intel i7) running Mac OS X and Windows.
- **Networking Instruction Lab:** Supporting networking course assignments, 2 racks with 8 Cisco 2911 routers and an assortment of network switches and Intel-based PC end-hosts.
- **Information Security Instruction Lab:** Supporting information security courses, a 16-seat cluster of SuperMicro workstations. Student teams are provided access to the latest information security hardware and software in an isolated environment allowing for study, analysis, and simulation of current threats without risk to production facilities.
- **Dune Instructional Storage Cluster:** Supporting HPC and large-scale data analysis courses, a 5-node GlusterFS storage cluster (2-socket, 6-core Xeon, 32 GB RAM each) providing over 500 TB of data storage with 10 Gbps IP network interface controllers and with high-speed QDR Infiniband to the Jinx HPC cluster.
- **Jinx Instructional HPC Cluster:** Supporting advanced programming courses, a 30-node, 336-core, GPU accelerated Torque/Maui cluster consisting of 24 HP Proliant SL390 servers and 6 Dell PowerEdge R710 servers. The HP servers have two Intel Xeon X5650 6-core processors and 24GB of RAM. Twelve of the HP servers are equipped with two NVIDIA Fermi-based Tesla M2090s and the other twelve with M2070s. The Dell

servers have two Intel Xeon X5570 4-core processors and 48 GB of RAM. The entire cluster is connected with QDR Infiniband to the Dune Storage Cluster and to the IP network with Gigabit Ethernet.

- **New HPC Cluster** (coming online Summer 2014): Supporting advanced programming courses, a 20-node, 640-core, compute cluster consisting of SuperMicro servers (2-socket, 16-core, 2.4GHz AMD Opertron 6378, 256GB RAM, 128GB SSD, QDR Infiniband to the Dune Storage Cluster, 10Gbps IP network cards).
- **Factor Instructional HPC Cluster:** Supporting operating and distributed systems courses, a 21-node, 216-core server cluster of Dell PowerEdge R610 servers (2-socket, 4-core, Intel X5550, 48GB RAM) and Dell PowerEdge R620 servers (2-socket, 6-core 2.0GHz Intel ES-2630L, 128GB RAM) with 2 Dell PowerEdge R710 file servers with 7TB of disk storage. The cluster resources are managed with Openstack.

Research Facilities

An abundance of research facilities are housed in the College's Schools and Research Centers:

- The School of Computational Science and Engineering (CSE) is located in KACB and supports substantial computational facilities related to both education and research. The School is affiliated with several research centers, initiatives and labs including the Institute for Data and High Performance Computing (IDH), the Keeneland project, the FODAVA Center, and the specialized High Performance Computing (HPC) Laboratory. Through industrial partnerships, the HPC Lab operates or supports several state-of-the-art parallel computers and future technologies, which are readily available for teaching and research and provide a diverse collection of resources for algorithmic exploration:

    - **Ion Cluster:** an 8-node, 64-core, GPU-accelerated Torque/Maui cluster consisting of Appro 1424x Twin-Servers (each with 2-socket, Intel X5550 4-core, 24GB RAM, QDR Infiniband, and 2 NVIDIA C1060 cards).
    - **Bugs Cluster:** a 6-node, 48-core cluster of Dell PowerEdge 1950 and 2950 servers (each with 2-socket, 4-core, Intel E5420 processors, 16GB RAM) configured with Hadoop.
    - **Convey FPGA servers:** The HPC Lab also utilizes two Convey HC-1 hybrid-core servers featuring field Programmable Gate Arrays (FPGAs) coupled with multi-core Intel Xeon processors.
    - **Topaz Cluster:** a 36-node, 288-core cluster of TeamHPC servers (each with 2-socket, 4-core processors).
    - **Sun Fire X4470 M2 large memory servers:** Two, dual socket, 8-core Intel Xeon E7-4820 processors with large memory configurations: 1TB RAM and 500GB RAM.
    - **Intel large memory server:** an Intel Server System QSSC-SR4 with four E7-8870 10-core Intel Xeon processors and 256 GB of RAM, the highest-ranked single-node system on the Graph500 benchmark.

    Additional CSE high performance computing resources include:

    - **Keeneland Full Scale (KFS) System** (installed in 2012)**:** a 120-node, 264 CPU and 360 GPU cluster, composed of HP Proliant SL-250 servers each with two Intel Sandy Bridge CPUs, 32 GB of host memory, 3 NVIDIA M2090 GPUs, and a Mellanox FDR InfiniBand interconnect.

- **Cray Supercomputers:** Through multiple projects and collaborations, the HPC Lab has access to massively multithreaded Cray XMT-series supercomputers. The Cray XMT series is similar to the Cray XT series supercomputers, but replaces the commodity x86 processors with unique latency tolerant processors that allow for fine-grained parallelism through 128 hardware thread contexts per processor. These processors scale memory bandwidth across multiple terabytes of RAM. The HPC Lab currently uses a Cray XMT located at Pacific Northwest National Laboratory with 128 processors and 1 TB of RAM as well as a next-generation Cray XMT2 at the Swiss National Supercomputing Center with 64 processors and 2 TB of RAM.
- **System Biology Center – Myriad Cluster:** a 10,000-core Penguin Computing cluster with a 100 TFLOP (teraflop) theoretical maximum performance, ranking within the top 100 supercomputers in the world.
- **FoRCE Research Computing Environment:** a Georgia Tech community resource that includes a mixture of compute nodes, some with attached GPUs, some with large memory capacity and some with local storage (56 total compute nodes, 1,592 total CPU cores).

- The **Georgia Tech Information Security Center (GTISC)** is located in KACB and is comprised of the Information Security Lab and the Converging Infrastructures Security (CISEC) Lab. GTISC operates a substantial number of network, computational server and storage resources to support its research activities in the area of information security.

- The **Networking and Telecommunications Group (NTG)** is located in KACB and includes the GT Network Operations and Internet Security Lab (GT Noise). NTG operates a substantial number of network, computational server and storage resources to support its research activities in the area of networking and security. The Noise lab also hosts a 70-node, Dell PowerEdge R410 compute cluster (840 CPU cores) as part of the multi-institute, VICCI programmable cloud-computing research testbed.

- The **Center for 21st Century Universities (C21U)** is located in KACB and operates the C21U Studio and Innovation Lab which includes a highly connected classroom, control room, and broadcast quality studio, as well as a dedicated support team to support experimental teaching and research in fundamental change in higher education.

- The **Center for Experimental Research in Computer Science (CERCS),** is located in KACB and includes the Interactive High-Performance Computing Lab (IHPCL), serving as a focus for interdisciplinary research and instruction involving high-performance computer systems. These facilities are linked by a dedicated high-performance backbone utilizing 10 Gbps Ethernet, and include:

  - **Whitestar Cluster:** an 840-node, 3360-core IBM BladeCenter LS20 cluster (2-socket, 2-core AMD Opteron 270, 4GB RAM each) configured with VMware VCenter.
  - **Jedi Cluster:** a 80-node, 760-core, Penguin Computing cluster with 30 Relion 1752 servers (2-socket, 6-core, 2.66GHz Intel X5650, 48GB RAM each) and 50 Relion 1702 servers (2-socket, 4-core, 2.4GHz Intel E5530, 24GB RAM each), configured with Openstack.

- **Maquis Cluster:** a 20-node, 160-core IBM BladeCenter H cluster (2 socket, 4-core, 1.86GHz Intel E5320 processors).
- **Vogue Cluster:** an 11-node, 88-core cluster of 4 Dell PowerEdge R610 servers (2-socket, 4-core, Intel E5550 processors, 12GB RAM) and 7 Penguin Computing Relion 1700 servers (2-socket, 4-core, Intel E5506 processors, 12GB RAM).
- **Rohan Cluster:** a 51-node, 102-core cluster of Dell PowerEdge 1850 Linux servers (2-socket, Intel Pentium4 Xeon EMT64 processers, Infiniband interconnects)
- **Polynesia/Samoa Cluster:** a 20-node, 180-core cluster of Dell PowerEdge 1950 servers (2-socket, 4-core, 2.5GHz Intel E5420 processors, 1GB RAM).
- Poster Printer (HP DesignJet 800).

- The Computer Architecture research group is located in KACB and conducts research on all aspects of future microprocessor technology including performance, power, multi-threading, chip-multiprocessing, security, programmability, reliability, interaction with compilers and software, and the impact of future technologies.

  - **Pasta Cluster:** a 35-node, 312-core cluster of 25 Dell PowerEdge 1950 servers (2-socket, 4-core, 3.0GHz Intel X5650 processors, 16GB RAM), and 2 Dell PowerEdge R410 (2-socket, 3GHz Intel X5450 processors, 24GB RAM) and an IBM BladeCenter HS22 (8-blades, 2-socket, 6-core, 2.93GHz Intel X5670 processors, 48GB RAM).
  - **Sushi Cluster:** a 14-node, 112-core cluster of 10 Dell PowerEdge 1950 servers (2-socket, 2-core, 3.0GHz Intel E5160 processors, 8GB RAM) and 4 Dell PowerEdge 1950 servers (2-socket, 4-core, 3.0GHz Intel E5450, 16GB RAM).

- The **GVU Center** is located in the TSRB Building and houses a variety of research labs in a multi-facility collection of workplaces. Total GVU lab space comprises more than 8,000 square feet. In addition, GVU affiliated laboratories are operated by non-CoC faculty in the College of Architecture; the School of Literature, Culture, and Communication; the School of Psychology; and the Interactive Media and Technology Center. GVU facilities utilize state-of-the-art high-performance servers and graphics workstations from major manufacturers such as Dell, HP, Apple and Sun. GVU is also a partner in the Aware Home Research Initiative (AHRI). A partial list of specialized GVU resources includes:

  - **The Aware Home:** a 3-story, 5,040 sq. ft. house and living laboratory for interdisciplinary research in design and social questions.
  - **The App Lab:** a "hackerspace" devoted to the creation of mobile applications and technologies across a range of platforms.
    - Numerous iOS, Android and Kindle devices for checkout.
    - Several dual-booted OSX and Windows workstations with current, mobile and gaming software dev environments, including Unity for publishing.
  - **The Proto Lab:** a 1,200 sq. ft. lab devoted to the prototyping of experimental devices such as wearable computers and equipped with devices such as:
    - 3D Printer (Dimension SST 768)
    - 3D Scanner
    - Laser Cutter/Etcher (Epilog)
    - Circuit Mill (LPKF ProtoMat S62)
    - CNC Router (K2CNC 4'x8')

        - Shop equipment (band saw, table saw, drill press)
        - Surface mount and through-hole soldering stations
        - Bench Equipment (Power Supplies, Oscilloscope/Logic Analyzer, RF Generator and spectrum analyzer)
        - Arduino Development/Test Circuit Boards
        - Silk Screening Equipment (for conductive ink)
        - Embroidery Machines (for conductive thread), sewing equipment, leather stitcher
    - **The Usability Lab:** complete with separated viewing area, for conducting and capturing video and screen recordings of computer-based studies.
    - A High-Definition (HD) Video Conferencing System (LifeSize).
    - A Video Webcasting AV Cart with High-Definition (HD) capture capability.
    - Sony Bloggie HD Cameras for field recording
    - 12 Camera IR Motion Capture System (Vicom)
    - Several Polhemus, Ascension, and Intersense tracking systems and head-mounted displays
    - Several Smartboards
    - A Segway Human Transporter
    - A Poster Printer (HP DesignJet 800)

- The **Institute for Robotics and Intelligent Machines (IRIM)** is located in the College of Computing Building and houses a variety of research labs in a multi-facility collection of workplaces. In addition, IRIM affiliated laboratories are operated by non-CoC faculty in the College of Architecture, College of Engineering: Schools of Aerospace Engineering, Biomedical Engineering, Mechanical Engineering, Electrical and Computer Engineering, College of Science: School of Physics, and the Georgia Tech Research Institute. A partial list of specialized IRIM and robotics equipment includes:

    - **Vehicles:**
        - 1 Porsche Cayenne outfitted for DARPA Grand Challenge competition.
        - 1 Actuated AM General Hummer (w/DGPS)
    - **Robots:**
        - 3 cells with KR210 KUKA robots, material handling equipment, and AGVs.
        - Several Kuka robotic arms
        - A Schunk robotic arm (LWA3) with dexterous hand (SDH2).
        - **Golem Krang:** a mobile manipulator designed and built by the Humanoid Robotics Lab, featuring a Schunk robotic arm mounted on a custom Segway Human Transporter.
        - **Simon:** a face-to-face, robotic research platform featuring an upper-torso humanoid social robot with two 7-DOF arms, two 4-DOF hands, and a socially expressive head and neck, including two 2-DOF ears with full RGB spectrum LEDs.
        - A Segway RMP200 Research Mobility Platform.
        - A Mobile Robotics PeopleBot.
        - A PR2 Willow Garage robot.
        - Rovio WowWee mobile webcam.
        - 18 Sony AIBO legged robots
        - 2 iRobot ATRV minis, 1 IS Robotics Pebbles III robots
        - 4 Pioneer 2-DXE, 3 Pioneer AT robots

        - 1 Evolution Scorpion, 1 Evolution ER1, 1 Segway, 1 Denning DRV-I robot
        - 3 RWI ATRV-Jr, 5 ActivMedia Amigobots, 1 Nomad 200, 5 Nomad 150, 1 Hermes II Hexipod, 3 Blizzard robots
        - several SICK scanners, various lasers, vision/motion systems, cameras, and associated equipment.
    - **Fabrication Shop:** a lab with band saws, drill presses, lathes, presses, grinders, etc. for the fabrication of robotic components.
    - **Electronics Shop:** a lab with oscilloscopes, logic analyzers, programmable power supplies, soldering equipment, etc.
    - **Wilks Cluster:** a 15-node, 180-core Supermicro 6016T-NTF compute cluster (2-socket, 6-core, 2.67GHz Intel 5650, 96GB RAM).
    - A Segway Human Transporter.
    - A Poster Printer (HP DesignJet 800).

Georgia Tech Research Networking Capabilities:

Georgia Tech's state-of-the-art network provides capabilities with few parallels in academia or industry, delivering unique and sustained competitive advantage to Georgia Tech faculty, students, and staff. Since the mid-80's Georgia Tech and OIT have provided instrumental leadership in high-performance networks for research and education (R&E) regionally, nationally, and internationally.

A founding member of Internet2 (I2) and National LambdaRail (NLR) – high bandwidth networks dedicated to the needs of the research and education community – Georgia Tech manages and operates Southern Crossroads (SoX, the I2 regional GigaPOP) and Southern Light Rail (SLR, the NLR regional aggregation). We work within six Southeastern states to make affordable high-performance network access and network services available to researchers and faculty at Georgia Tech, their collaborators, other higher-education systems, K-12 systems, and beyond.

Georgia Tech's network has high-performance connectivity to other members of the research and education community world-wide through dual 10 Gbps (gigabits per second) links to SoX/SLR, which is peered with NLR Packetnet, Internet2 Network, TransitRail, Oak Ridge National Labs (ORNL), the Department of Energy's Energy Sciences Network (ESNet), NCREN, NASA's NREN, MREN, FLR, Peachnet, LONI, 3ROX, as well as other SoX participants in the Southeast.

In addition to the exceptional R&E network connectivity provided to all Georgia Tech faculty, students, and staff, dedicated bandwidth in support of specific collaborations and research is also possible.

Facilities, Equipment & Other Resources

Laboratory: The Systems and Software Security (S^3) lab, founded and directed by the PI Lin, is located in ECSS 3.612 at UT Dallas. It has over 300 square feet of laboratory space that can host 10 PhD and MS students.

Clinical: N/A

Animal: N/A

Computer: The S^3 lab is currently equipped with **Six** Dell Optiplex 790 desktops, each with Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8G RAM, 1T Disk; **Two** Dell T7500 servers. One with Intel(R) Xeon(R) CPU E5620 @ 2.40GHz, 24G RAM, and 2T Disk; the other with 48G RAM.

Office: The PI has 120 square feet of office space equipped with **one** PC with Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, 8G RAM, 1T Disk;

Other Resources: The Department of Computer Science at UT Dallas contain several computer clusters for use by students and researchers. In addition, the department has a Security Analysis and Information Assurance Laboratory (SAIAL), a $1.5 million state-of-the-art laboratory. The lab consists of three separate rooms each individually tested to meet MIL-STD-285 TEMPEST standards. This lab is currently used primarily for Cloud Computing and Digital Forensics research projects.

Data Management Plan

Following the NSF Data Management Guidelines, we will administrate, archive, and maintain the data and the intellectual properties generated during this project. In this document, we describe the expected data that we will produce and collect, and a detail plan to openly manage them without violating user's privacy over the course of the project.

I. Expected Data. This proposed work is expected to generate new techniques, develop new systems and software tools, and have new curriculum materials and hands-on labs. In general, the data to be produced by the proposed research falls into the following categories:

1. **Software Prototype** includes source code of SGX infrastructure, toolchains and kernel extension, as well as the tools to secure SGX program and defend SGX malware.
2. **Experimental Data and Result** that characterizes the behavior of our prototype, and includes the details of experiment settings and analysis results, and the security benchmark suite to evaluate SGX programs.
3. **Academic Papers and Technical Reports** that describe the work proposed in this research.
4. **Curriculum Materials** including lecture notes, presentation slides, homework and solutions, projects, in systems security and hands-on labs.

II. Management Plan. As our goal is to make an tangible influence to our society, all the developed techniques and implementation, will be maintained publicly accessible, and tech-transfer to the open source team (e.g., Apache organization) for future maintenance. More specifically, in accordance with NSF data sharing policy for scientific research to ensure its usability, accessibility, and preservation, we use the following plan to manage each specific category of data above:

1. **Experimental Data**. The input and output data conducted to evaluate our prototype will be archived by the PI for the duration of the project and made public available. In addition, the source code and experimental data will also be permanently hosted as a digital archive operated by Georgia Tech Information Security Center (GTISC) and backed up by the D-Space system at University of Texas at Dallas. The operational team at GTISC is a group of permanent system administrators who are specialized in maintaining our group's infrastructure. The central repository (e.g., gitolite, code review system, monitoring, and backup services) is protected with RAID5 in local storage, and is backed up daily to a dedicated archiving server.
2. **Software and the Source Code**. For prototype that is built on top of open source project (e.g., GPL) will be correspondingly licensed using the same policy. If the software is entirely developed from scratch without using any other proprietary software, PIs will make it open source with detailed documentation using Apache license. In particular, we plan to make an early release of our research prototype via our group's GitHub repository at https://github.com/sslab-gatech/, and https://github.com/utds3lab. For instance, our preliminary system OpenSGX is already publicly available at https://github.com/sslab-gatech/opensgx.
3. **Academic papers**. Academic conference and journal papers, technical reports, presentation slides, posters, and other materials will be released through PIs' web site (e.g., https://sslab.gtisc.gatech.edu/. While published papers are subject to the discretion of copyright owners (e.g., ACM, IEEE, USENIX), the technical reports will be made freely available without any restriction.
4. **Curriculum Materials**. Curriculum materials will be openly shared with anyone in the world through PIs' web site. Except homework solutions and hands-on labs require permissions from instructors, all other materials is freely available.

Mentoring Plan

Not applicable

Collaboration Plan

This proposed research will be conducted by two investigators from Georgia Tech and UT Dallas with a wide range of expertise from operating systems, security, cloud computing, programming languages and compilers. Both PIs share strong interests in system security, and more specifically have unique expertise to the following topics in order to complete all aspects of the proposed work:

- **PI Kim:** Operating system design, security, programming languages, manycore operating systems, and computer architecture.
- **PI Lin:** Virtualization, systems security, binary code analysis, compilers, and cloud computing.

Past Collaboration. The two investigators have established collaboration recently. They have ongoing collaborations on several fronts including the design of new security applications with SGX. These collaborations so far have helped bridge the different specialties resulting in co-advising students, and design and implementation of new tools and systems. Also, the joint publications will come soon.

Joint Workshop Organization. Both PIs participated the International Security Education Workshop that GTISC, NSF and Intel jointly organized at Georgia Tech in May 2015. Through the educational workshop, we have confirmed a huge interests and demands, not only from academic institutions but also from industries and government. In addition, both PIs participated the Intel SGX Workshop held as a part of ISCA 2015, and shared our early experiences and potential research directions with Intel researchers and people in academia. Based on feedback and our early experiences, we are planing to organize a joint workshop on Intel SGX in one of security conferences in near future.

Online Community. To foster open source community, our team will maintain a forum site to share our experiences of building SGX applications. Since PI Kim leads the OpenSGX project and already maintains it as an open source project, our team, as part of this proposed work, extends it to broader communities by creating a community forum and by providing tutorials and better guide line.

Collaboration Mechanisms. To have a successful collaboration and complete the proposed project on time, we plan to use the following collaboration mechanism:

1. *Regular Meetings:* The project team including the students will have a bi-weekly research meeting through the web and teleconferencing. The investigators already use such tools (Skype, Webex, Google docs) for remote collaborations in their ongoing collaborative projects. Further, we will have quarterly day-long web-based technical presentations to share critical progress and identify significant road blocks. These presentations will also enhance the communication skills of the graduate and undergraduate students involved in the project. An annual day-long teleconferencing meeting (or near security conferences) will also be conducted for an annual assessment of the research progress. We will invite our industry collaborators from Intel to join both our quarterly and annual meeting, interact with them and also hear their feedbacks.
2. *Student Exchange:* Student exchange between two campuses will further foster collaboration and interaction among the students who participate in the projects. We will perform this exchange once throughout the projects in a rotating fashion among the students.
3. *Industry Input:* We will interact with our industry collaborators in particular Intel for their input on the technology front. This will be accomplished through internship opportunities for our students, having industry collaborators on student thesis committees and through co-authorship of research articles.
4. *Web-based Interaction:* We will maintain a central repository for the project at Georgia Tech. This will have pointers to all major developments, papers, technical reports, tools, and education and outreach activities. This repository will also contain extensive simulation data/traces to help outside researchers, as well as links to related projects to facilitate interaction with different research groups worldwide.

List of PI Kim's Recent Collaborators

1. Thomas Anderson (University of Washington)
2. Alexandra Boldyreva (Georgia Tech)
3. Soham Desai (Intel)
4. Young Ik Eom (Sungkyunkwan University)
5. Hadi Esmaeilzadeh (Georgia Tech)
6. Dongsu Han (KAIST)
7. Bill Harris (Georgia Tech)
8. Prerit Jain (Oracle)
9. M. Frans Kaashoek (MIT)
10. Brent Kang (KAIST)
11. Yongdae Kim (KAIST)
12. Arvind Krishnamurthy (University of Washington)
13. Sang-Won Lee (Sungkyunkwan University)
14. Wenke Lee (Georgia Tech)
15. Long Lu (SUNY)
16. Gloria Mainar-Ruiz (Microsoft Research)
17. Robert Morris (MIT)
18. Todd C Mowry (CMU)
19. Onur Mutlu (CMU)
20. Mayur Naik (Georgia Tech)
21. Marcus Peinado (Microsoft Research)
22. Tielei Wang (Georgia Tech)
23. Xi Wang (University of Washington)
24. David Wetherall (Google)
25. Xinyu Xing (Pennsylvania State University)
26. Nickolai Zeldovich (MIT)

List of PI Lin's Recent Collaborators

1. David Brumley (Carnegie Mellon University)
2. Juan Caballero (IMDEA)
3. Haibo Chen (SJTU)
4. Guofei Gu (Texas A&M)
5. Kevin Hamlen (UT Dallas)
6. Xuxian Jiang (Qihoo)
7. Murat Kantarcioglu (UT Dallas)
8. Ashish Kundu (IBM Research)
9. Latifur Lkhan (UT Dallas)
10. Charles McFarland (Intel)
11. Junghwan Rhee (NEC Research Lab)
12. Ryan Riley (Qatar University)
13. Kevin Roundy (Symantec Research Lab)
14. Weidong Shi (University of Houston)
15. Bhavani Thuraisingham (UT Dallas)
16. Dongyan Xu (Purdue)
17. Shouhuai Xu (UT San Antonio)
18. Heng Yin (Syracuse)
19. Junyuan Zeng (FireEye)
20. Mingwei Zhang (Intel)
21. Xiangyu Zhang (Purdue)

List of Project Personnel and Partner Institutions

As explained and justified in the main proposal, if funded this project will be conducted in collaboration between the Georgia Institute of Technology and The University of Texas at Dallas. More specifically:

1. **Dr. Taesoo Kim**, Georgia Institute of Technology, PI.
2. **Dr. Zhiqiang Lin**, The University of Texas at Dallas, PI.

Other Partner Institutions: **Intel Corporation**. The PIs will also be engaged with the SGX researchers and engineers at Intel. Details on the collaboration plan with Intel is presented in the attached letter.

September 13, 2015

Program Director
National Science Foundation (NSF)
Washington DC, USA

Dear NSF Program Director,

I am a senior researcher in the Office of the CTO at Intel Security. My research has been focusing on building the use case of Intel® Software Guard Extension (SGX), and analyzing the potential malware that uses Intel® SGX since 2013. I am also the technical point of contact of Dr. Lin's Intel® SGX malware analysis project supported by Intel®.

I am pleased to write this letter elaborating my collaboration commitment with Dr. Kim and Dr. Lin on their proposed SGX research. Since August 2015, SGX has been publically available, enabling new hardware level support in securing both data and instructions from unauthorized access. Previously, securing code against users heavily relied on trust of the operating system platform. By enabling the use of secure enclaves within the CPU instruction set, the operating system no longer needs to be trusted to maintain integrity of code execution. Intel® SGX provides the ability to secure code and data but further research into practical applications, special considerations, new security concerns and new software development models is needed. I am excited to learn that Dr. Kim and Dr. Lin are proposing developing open source "systems, tools and techniques for executing and securing SGX programs", which also gives me and my team members an opportunity of collaborating with the two PIs on some of their proposed research tasks.

Specifically we would be very much interested in being the first set of users to test all of the developed prototypes including their systems support, the tool chains, the sgxlib, and SGX execution sandbox. Second, my team and I would like to continue collaborating with Dr. Lin on SGX malware analysis. In particular, we would like to contribute our expertise and work together with the two PIs on extracting the best set of available features that can characterize the enclave program execution, an open problem to be solved in SGX program execution. Finally, of particular note is that I like the new proposed SGX program execution model that requires explicit system

Intel Corporation
5000 Headquarters Drive
Plano Texas 75024

call specification. I believe this is a practical SGX execution model, and I would like to be engaged deeply in this proposed task.

To conduct this research, our Intel researchers including me plan to meet regularly with the PIs and their students, to discuss the technical details, and evaluate the proposed systems and tools.

I believe Dr. Kim and Dr. Lin's proposed project is very useful and may have significant impact on SGX software development. If you have any questions, please feel free to contact me.

Sincerely,

Charles McFarland
Senior MTIS Research Engineer
Office of the CTO
charles.mcfarland@intel.com
(972) 963-7140

Intel Corporation
5000 Headquarters Drive
Plano Texas 75024

Topic Areas

Primary topic area:

1. Systems
2. Software
3. Trust

Secondary topic area:

1. Cloud
2. Hardware
3. Privacy, applied