THE RESEARCH AND METHODOLOGY DIRECTORATE

Application Level Cryptography for Securing Online Survey Responses

Simson L. Garfinkel and William Yates

February 2019

United States® Census Bureau

census.gov

Application Level Cryptography for Securing Online Survey Responses

Simson L. Garfinkel and William Yates

February 28, 2019

1 Introduction

The Census Bureau and other U.S. statistical agencies increasingly rely on internet-based self-response instruments to collect data from individuals and establishments. Because internet-connected computers have long been under attack by both cyber criminals and foreign governments[1], these connections are now subject to monitoring by the U.S. Department of Homeland Security (DHS).

Respondent data submitted to the Census Bureau website is protected using the Transport Layer Security (TLS)[2] cryptographic protocol and cannot be deciphered by DHS as the DHS monitoring system is currently deployed.

Because the DHS system might change in the future in response to new threats, the Census Bureau has developed and demonstrated an approach that uses a second layer of encryption to protect respondent data. This system would prevent DHS employees from accessing respondent data even if DHS were to be provided with the Census Bureau's TLS encryption keys.

1.1 Background

The Census Bureau and other U.S. statistical agencies collect data from respondents under a pledge of confidentiality which states that data collected will be used for statistical purposes only. In particular, Title 13 of the U.S. Code prohibits information that the Census Bureau collects from being used for law enforcement purposes.

Statistical agencies are increasingly accepting data from respondents using internet self-response instruments. The Census Bureau plans to make heavy use of internet self-response for the 2020 census.[3] At the same time, computers operated by the U.S. Government are under constant attack. A successful attack on Census Bureau computers would also represent a significant threat to data confidentiality. For this reason, the Census Bureau, like other U.S. Government civilian agencies, participates in the EINSTEIN network monitoring program operated by the U.S. Department of Homeland Security.[4]

At the 13th Biennial Federal Committee on Statistical Methodology (FCSM) Policy Conference, a session explored "The Challenges of Overlapping Mandates between Federal Statistical Agencies and Departmental Chief Information Officers."[5] One of the panel speakers was Wayne R. Smith, the former Chief Statistician of Canada and Head of Statistics Canada, who resigned his position in September 2016 over concerns that Statistics Canada's move to shared information services meant that the agency would be unable to protect the confidentiality of respondent data[6]. After Dr. Smith spoke, several attendees expressed concern that the U.S. Government's "EINSTEIN" program might pose a similar threat to confidentiality for U.S. statistical agencies.

1.2 EINSTEIN

The EINSTEIN program, operated by the U.S. Department of Homeland Security (DHS), is designed to provide real-time collection, analysis, and sharing of computer security information across the federal government civilian agencies to help mitigate internet-based threats.[7]

The original EINSTEIN program was developed as part of the U.S. Governments' Trusted Internet Connection (TIC) program, an outgrowth of the 2003 National Strategy to Secure Cyberspace.[8] The original EINSTEIN system was designed to monitor and record network flow records between federal civilian executive branch agencies and the public internet to provide after-action forensic analysis and support. The EINSTEIN 2 program, first deployed in 2008, incorporated an intrusion detection system to detect hostile activity against federal agencies in real time and provide notification to the victims. Four years later, DHS began transitioning to the EINSTEIN 3 Accelerated (E^3A) program, with the goal of both detecting and preventing cyber attacks against federal civilian government networks.[9]

Designed to be operated by the DHS Office of Cybersecurity and Communications (CS&C), EINSTEIN 3's feature set included deep packet inspection, in which the content of network streams are examined by the system.

In April 2013, the Department of Homeland Security published *Privacy Impact Assessment for EINSTEIN 3 — Accelerated (E^3A)*[10]. According to the PIA, "E^3A combines existing CS&C analysis of EINSTEIN 1 and EINSTEIN 2 data as well as information provided by cyber mission partners with existing commercial intrusion prevention security services to allow for the near real-time deep packet inspection of federal network traffic to identify and react to known or suspected cyber threats." [10, p.4]

The PIA explained that while deep-packet inspection might occasionally result in the EINSTEIN 3 sensors encountering personally identifiable information (PII), such information would be immediately deleted if it was captured (see Relationship Between Participants — Privacy Considerations on page 15). The PIA also stated that it might be necessary, at times, to include PII captured by EINSTEIN in an analytical product that DHS would then distribute. The PIA further stated that non-cybersecurity information collected by DHS might be disseminated for non-cybersecurity purposes (see Privacy Impact Analysis: Related to Information Sharing on page 15).

On May 6, 2016, DHS issued an update to the E^3A PIA, stating that E^3A would be enhanced with a new feature called Web Content Filtering (WCF) that would provide the ability to decrypt the Secure Socket Layer (SSL) protocol (see Reason for the PIA Update — Web Content Filtering). Note that SSL is the previous name for the TLS protocol that is used to protect World Wide Web traffic.

Consistent with these stated capability improvements in the EINSTEIN system, on December 14, 2016, the Census Bureau published a Notice in the Federal Register indicating its intent to revise the Census Bureau's confidentiality pledge.[11] That new pledge has since been adopted. It reads:

> "The U.S. Census Bureau is required by law to protect your information. The Census Bureau is not permitted to publicly release your responses in a way that could identify you. Per the Federal Cybersecurity Enhancement Act of 2015, your data are protected from cybersecurity risks through screening of the systems that transmit your data."[11,12]

1.3 Web Content Filtering of Encrypted Content

Generally speaking, there are two ways that modern web browsers can communicate with web servers over the internet: by sending Hypertext Transfer Protocol (HTTP) commands over the internet without encryption, and by wrapping HTTP commands within a TLS connection, sometimes called a TLS tunnel..

When a web page is accessed using a Uniform Resource Locator (URL) that begins with *http://* — for example, `http://census.gov` — the connection between the server and the browser is not

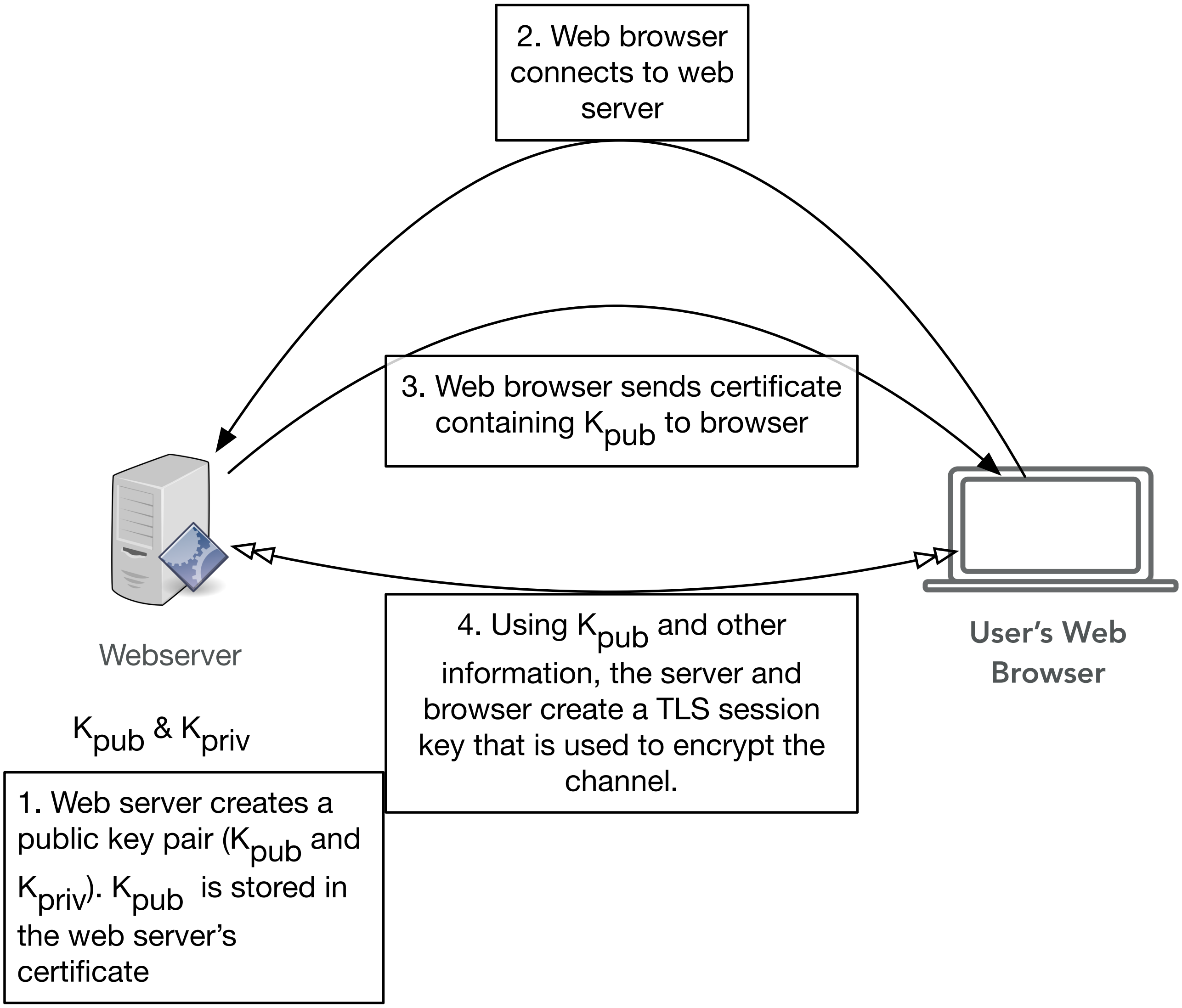

2. Web browser connects to web server
3. Web browser sends certificate containing K_{pub} to browser
Webserver
User's Web Browser
4. Using K_{pub} and other information, the server and browser create a TLS session key that is used to encrypt the channel.
K_{pub} & K_{priv}
1. Web server creates a public key pair (K_{pub} and K_{priv}). K_{pub} is stored in the web server's certificate

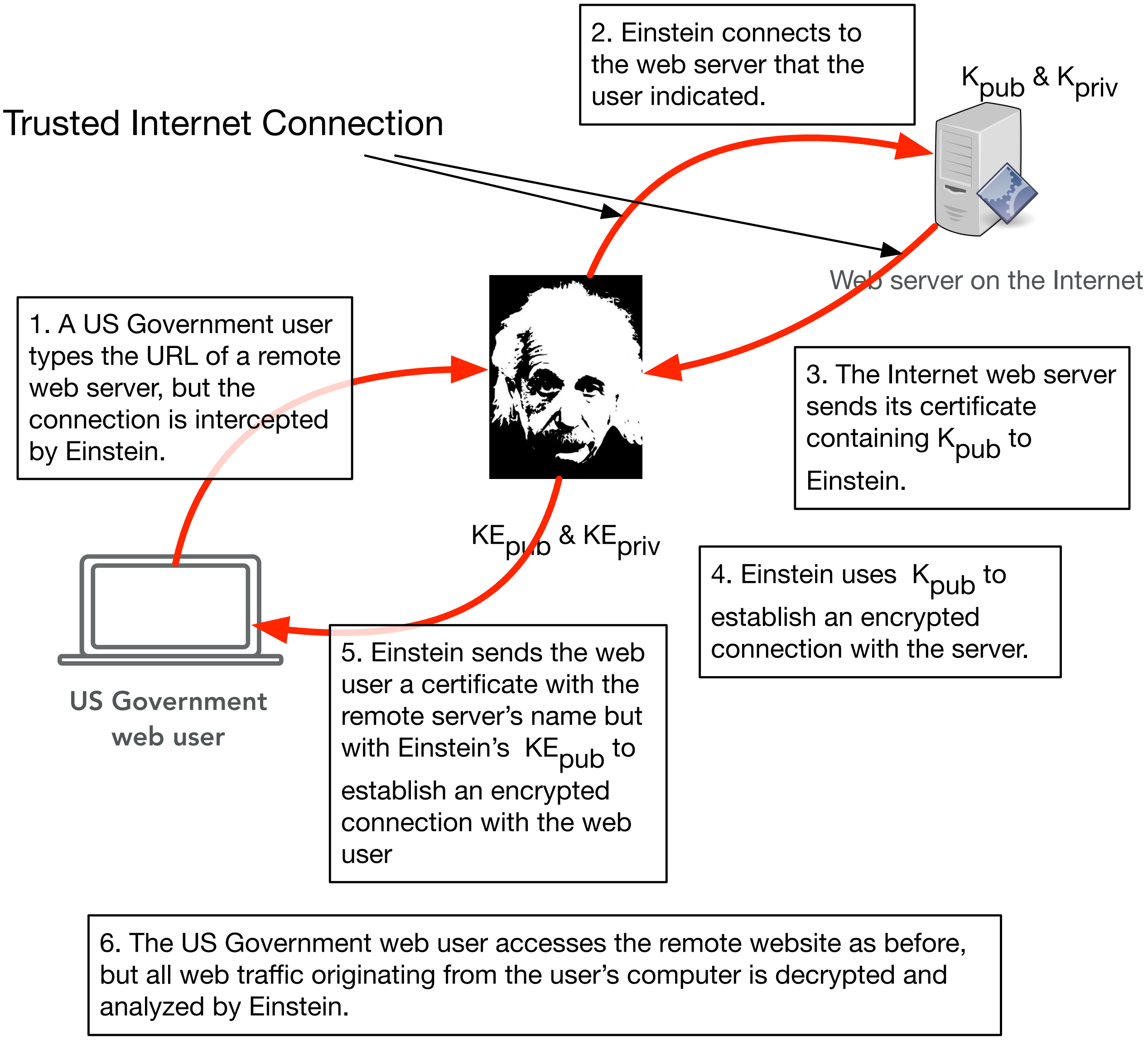
Figure 1. The Transport Layer Security (TLS) protocol, in brief

encrypted. Alternatively, when the same web page is accessed using a URL that begins with *https://* (e.g. `https://census.gov`), then the server and the browser exchange information over an encrypted connection.

By design, TLS provides for both confidentiality and integrity of the connection: it assures that a third party cannot eavesdrop on the contents of the connection, and the data sent by the browser is received by the server without modification.

TLS is based on public key cryptography. Each web server is configured with a private key and a public key. Information encrypted with the public key can only be decrypted using the corresponding private key. The web server's public key and its hostname are stored in its server certificate (Figure 1, step 1). When the web browser connects to the web server (step 2), the server sends to the browser the servers' certificate, which contains the public key (step 3). The browser and the server then create a session key that is used to encrypt the TLS connection (step 4).

Many commercial network appliances now offer a feature variously called "Deep Packet Inspection



1. A US Government user types the URL of a remote web server, but the connection is intercepted by Einstein.

2. Einstein connects to the web server that the user indicated.

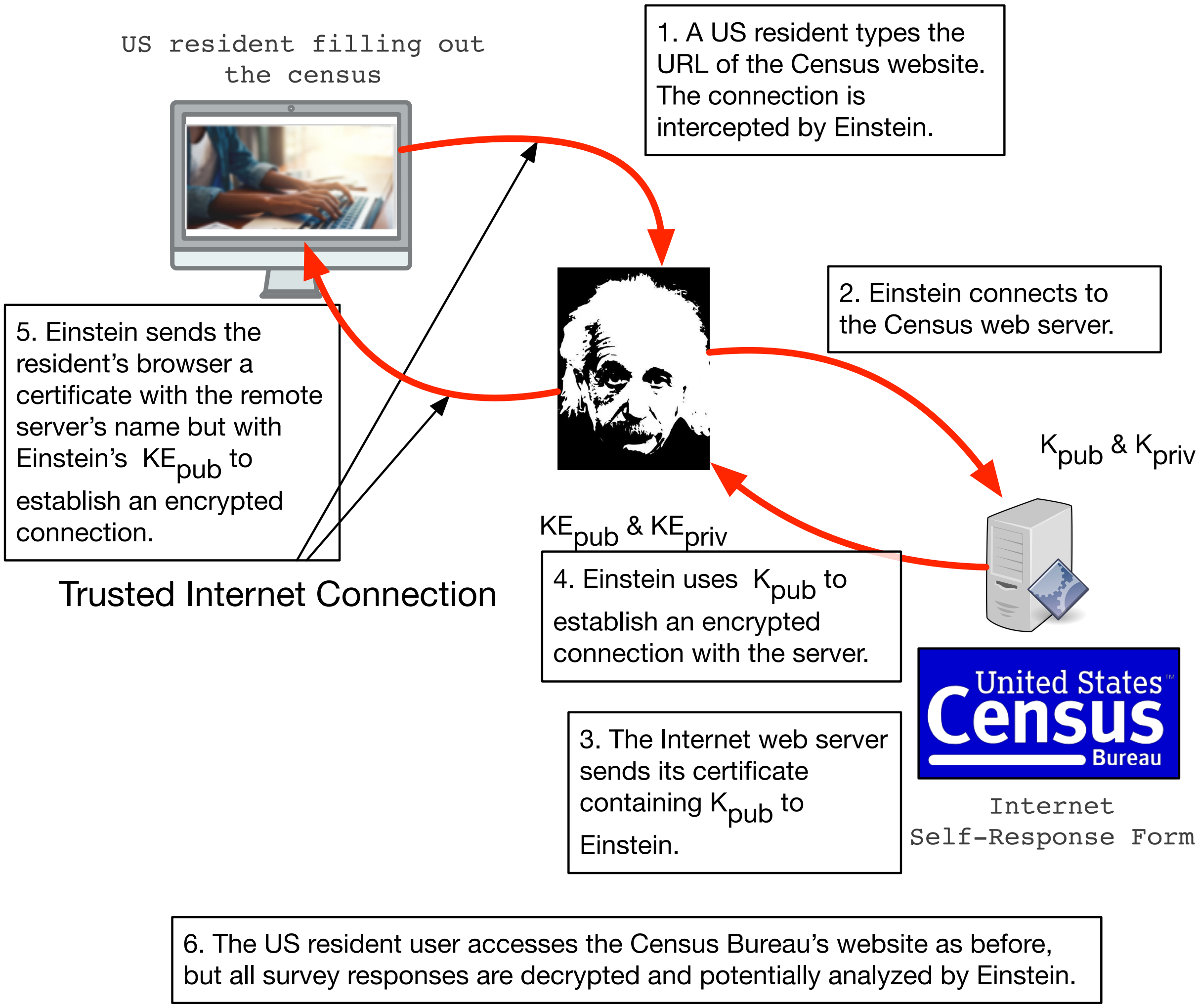
3. The Internet web server sends its certificate containing K_{pub} to Einstein.

4. Einstein uses K_{pub} to establish an encrypted connection with the server.

5. Einstein sends the web user a certificate with the remote server's name but with Einstein's KE_{pub} to establish an encrypted connection with the web user

6. The US Government web user accesses the remote website as before, but all web traffic originating from the user's computer is decrypted and analyzed by Einstein.

Figure 2. EINSTEIN 3A's TLS interception capabilities as envisioned for monitoring outbound web connections of US Government employees.


US resident filling out the census
Trusted Internet Connection
KE_{pub} & KE_{priv}
K_{pub} & K_{priv}
United States Census Bureau
Internet Self-Response Form

1. A US resident types the URL of the Census website. The connection is intercepted by Einstein.

2. Einstein connects to the Census web server.

3. The Internet web server sends its certificate containing K_{pub} to Einstein.

4. Einstein uses K_{pub} to establish an encrypted connection with the server.

5. Einstein sends the resident's browser a certificate with the remote server's name but with Einstein's KE_{pub} to establish an encrypted connection.

6. The US resident user accesses the Census Bureau's website as before, but all survey responses are decrypted and potentially analyzed by Einstein.

Figure 3. EINSTEIN 3A's TLS interception could also be used to monitor the Census Bureau's internet self response instrument.

of SSL/TLS Encrypted Traffic”[13], “TLS Decryption,”[14] and “SSL analysis,”[15]. These devices typically operate as a cryptographic proxy, essentially mounting a relay attack on the encrypted communications. TLS can’t prevent this sort of attack, but it allows the web browser to at least detect the attack, because the certificate provided by the interception is not issued by a trusted certificate authority.

The Web Content Filtering part of the DHS EINSTEIN 3A system is designed to eavesdrop on the TLS-protected connection (Figure 2). When a web user at a U.S. Government facility types the URL of a remote web server (step 1), the browser opens a connection not to the remote webserver, but to EINSTEIN 3A. It is the EINSTEIN system that then makes the connection to the remote web server (step 2). The remote web server sends its certificate back to EINSTEIN (step 3), and the two systems create a cryptographically protected channel that is used for future communication. Finally, the EINSTEIN system sends its public key to the web user and creates a second protected channel between EINSTEIN and the user (step 5). The browser’s warning of an untrusted certificate can be overcome by making the appliance a certificate authority and embedding the appliance’s certificate in the trusted certificate database of the web user’s computer.

TLS interception systems similar to EINSTEIN are now widely used on the internet. A 2016 study by O’Neill et al. found that 1 in 250 TLS connections on the internet were intercepted in this *proxied* manner. “The majority of these proxies appear to be benevolent, however we identify over 1,000 cases where three malware products are using this technology nefariously,” the authors concluded[16]. There is also evidence that foreign governments may have used this approach to spy on their own citizens[17].

There are many reasons that an organization might legitimately wish to monitor the content of TLS connections. For example, the organization could detect if a remote website was exploiting web browser vulnerabilities to inject malware. TLS interception can also help identify the command-and-control channels used by malware operating within an organization, helping an organization to identify and eliminate such electronic intrusions. Finally, TLS interception can be used to detect inappropriate network activity by employees.

TLS interception systems can also be turned around, and used to monitor inbound TLS connections to an organization’s web server originating on the public internet (Figure 3). In this configuration, the monitoring system is given a copy web server’s certificate and private key, so that remote web users are not aware that they are connecting to the monitoring device, rather than to the actual website. (TLS connections can be monitored if the packets are captured and later decrypted using the web server’s private keys, although this approach does not work for TLS version 1.3, which offers perfect forward secrecy.)

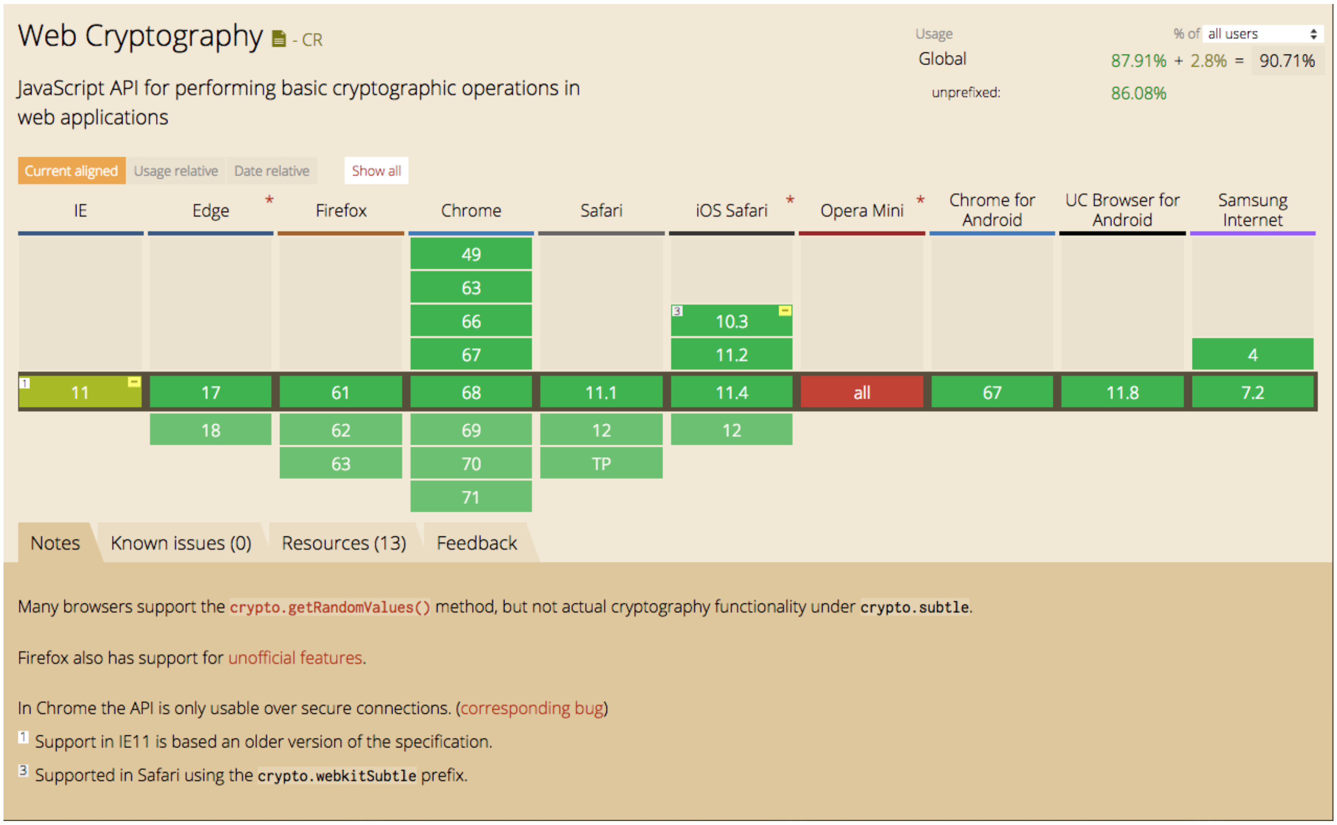
Given the widespread availability, use, and apparent acceptance of TLS decryption technology, we sought to develop a system that would protect respondent data even if the Census Bureau website was being monitored with technology that decrypted TLS-protected communications.

1.4 Securing Respondent Data With the Web Cryptography API

The World Wide Web Consortium’s Web Cryptography API[18] defines a set of cryptographic functions that can be used by JavaScript applications running within modern web browsers. According to the standard, typical uses for this technology are to enable Multi-factor Authentication, to encrypt documents locally that are then stored encrypted on a remote website, to sign documents, and to implement secure messaging. According to the Can I Use website, the Web Cryptography API is well supported in all current web browsers except for Internet Explorer 11 (Figure 4).

The Web Cryptography standard requires that web applications using Web Cryptography API be loaded into a web browser using the TLS protocol, in order to assure that an attacker does not modify the JavaScript application as it is being sent to the browser.

The Census Bureau has developed an approach that uses the Web Cryptography API to encrypt survey



Web Cryptography - CR

JavaScript API for performing basic cryptographic operations in web applications

Usage % of all users

Global 87.91% + 2.8% = 90.71%

unprefixed: 86.08%

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			66		[3] 10.3				
			67		11.2				4
[1] 11	17	61	68	11.1	11.4	all	67	11.8	7.2
	18	62	69	12	12				
		63	70	TP					
			71						

Many browsers support the `crypto.getRandomValues()` method, but not actual cryptography functionality under `crypto.subtle`.

Firefox also has support for unofficial features.

In Chrome the API is only usable over secure connections. (corresponding bug)

[1] Support in IE11 is based an older version of the specification.

[3] Supported in Safari using the `crypto.webkitSubtle` prefix.

Figure 4. Support for the Web Cryptography API as of September 2018, according to https://caniuse.com/#feat=cryptography.

responses sent from the web browser to the web server separately from the underlying HTTPS technology used to encrypt the web page itself (Figure 5). This second layer of encryption operates inside the HTTPS encrypted tunnel at the application data level. The process begins with the web server at the statistical agency creating a second encryption public key (Step 1) and used this second key to encrypt respondent data, then sent the respondent data to the web server for decryption. As a result, even if a TLS decryption appliance was used to decrypt the HTTPS stream, it would not have access to survey responses.

1.5 DHS Reactions

On October 13, 2017, a representative from the Census Bureau met with representatives from the Department of Homeland Security's National Cybersecurity Protection System (NCPS) to determine if the application-level encryption system could be deployed without violating any DHS policy and intentions[19]. At the meeting, DHS described the operation of the EINSTEIN system and the Census Bureau described the application-level encryption approach. DHS stated that deploying the application-level cryptography system would not violate any DHS policy, and that the system would provide additional security for Census Bureau-collected data, by allowing the data to reside in encrypted form on the Bureau's web server before it transitioned to a more secure system.

At the meeting, DHS also stated that in its current deployment, the E^3A Web Content Filtering was not designed to filter data sent from external web browsers to Government-operated web servers (e.g. Figure 3). Instead, E^3A was solely designed to monitor data sent between external websites and web browsers operated within the government networks (e.g. Figure 2). For example, E^3A could intercept data encrypted by malware running inside a user's web browser and communicating with external web servers.

Following the October meeting, the Census Bureau arranged for DHS NCPS to brief the Federal Committee on Statistical Methodology's Confidentiality and Data Access subcommittee (which took place on January 23, 2018), as well as the FCSM executive committee, to explain that E^3A would not be deployed in a way that could decrypt respondent data.

2 The Application Level Protection Mechanism

This section describes the application level cryptography mechanism.

2.1 Overview

A typical internet self-response survey consists of three parts:

1. A web application server, which sends the survey form to the respondent, and receives the respondent data from the respondent's web browser.

2. The survey form, which consists of a Hyper Text Markup Language (HTML) web page and associated JavaScript application that runs inside the respondent's web browser. The application presents the survey to the respondent, receives the data from the user, performs local data validation, and sends the data to the web application server.

3. A database connected to the web application server, which stores responses.

2.2 Prototype

In our development, we created two distinct prototypes.

The first prototype version encrypted each field of the survey form using the plain RSA encryption using the PKCS1 v1.5 algorithm. This version of RSA encryption uses the OAEP-based encryption

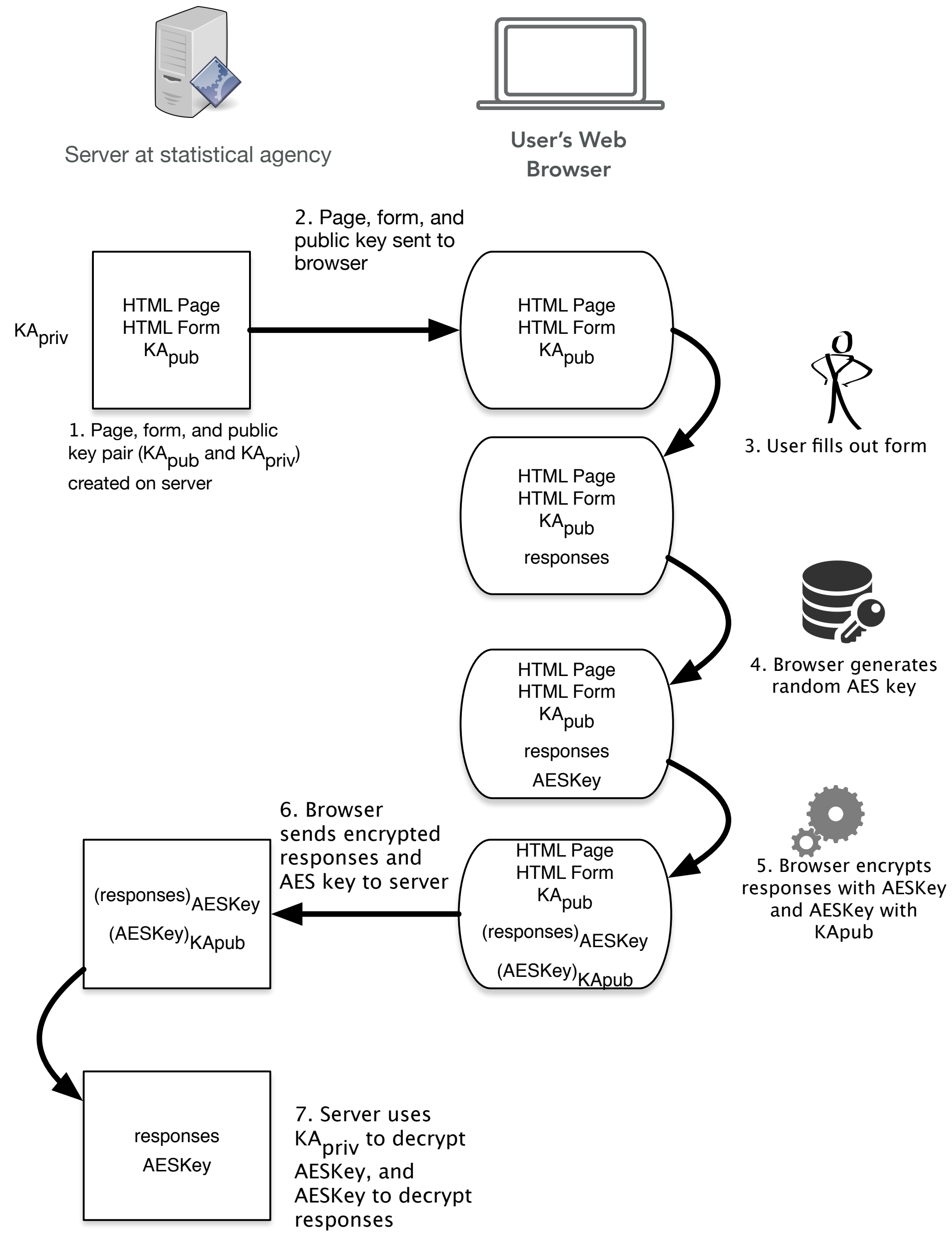

Server at statistical agency
User's Web Browser
2. Page, form, and public key sent to browser
KApriv
HTML Page
HTML Form
KApub
1. Page, form, and public key pair (KApub and KApriv) created on server
HTML Page
HTML Form
KApub
3. User fills out form
HTML Page
HTML Form
KApub
responses
4. Browser generates random AES key
HTML Page
HTML Form
KApub
responses
AESKey
6. Browser sends encrypted responses and AES key to server
(responses)AESKey
(AESKey)KApub
HTML Page
HTML Form
KApub
(responses)AESKey
(AESKey)KApub
5. Browser encrypts responses with AESKey and AESKey with KApub
responses
AESKey
7. Server uses KApriv to decrypt AESKey, and AESKey to decrypt responses

Figure 5. Application-level cryptography uses a second public/private key pair to encrypt survey responses.

scheme to protect against chosen plaintext and replay attacks. The prototype is conceptually easy to understand, but it is not very efficient.

Our second prototype generated an AES-256 session key and used this key to encrypt the contents of a JavaScript Object Notation (JSON) data structure that contained the survey responses. The RSA public key received from the web server was then used to encrypt the AES-256 session key. The encrypted session key and the encrypted JSON data structure are then sent together to the web server as a single package. This version is more efficient, as the length of the combined encrypted JSON data structure and a single RSA encryption is considerably smaller the size of encrypting each value with RSA.

Both prototypes implement a client/server system that displays a simple survey, encrypts the survey results in the user's web browser, and sends the encrypted results to the web server, where the encrypted survey responses are stored in a database. A second interface, created for a demonstration, shows both the encrypted and decrypted values.

Our prototype system modified the typical internet survey system in the following ways:

1. The web application server was modified to create a second public/private key pair that is used to encrypt respondent data (Figure 5, step 1). We call this the application-level key pair. In our implementation, the server generates a 1024-bit RSA key pair that is used to secure all respondent data.

2. We modified the HTML and JavaScript application sent from web application server to the browser in three ways. First, the application-level public key is embedded in the JavaScript application. (step 2)

   The user fills out the HTML page as before (step 3).

3. We added additional JavaScript code that creates a 256-bit AES session encryption key to encrypt the respondent data when the user clicks the "SUBMIT" button (step 4). This session encryption key is then encrypted with the application-level public key (step 5).

   The encrypted session key and the encrypted respondent data (the "encrypted package") are sent to the application server (step 6).

4. We further modified the application server to store the encrypted response in the database.

5. Finally, we created a new program for extracting the encrypted data from the database and decrypting it. Decryption is performed using the application-level private key to decrypt the 256-bit AES session key, which is then used to decrypt the respondent data (step 7).

2.3 Demo

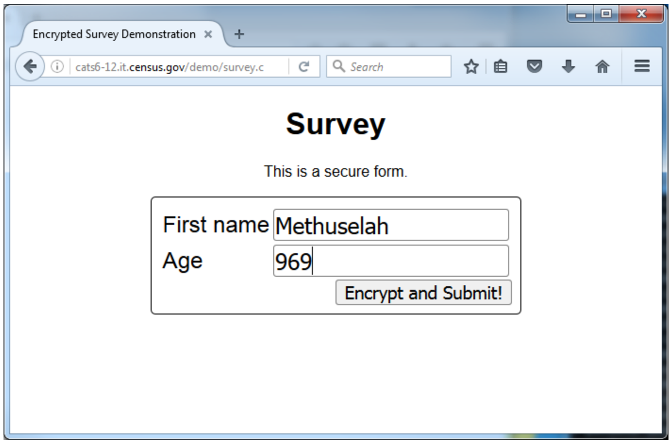
We create a simple web form for a hypothetical survey that requested a first name and an age (Figure 6). When submitted, these two values were packed into a JSON data structure (Figure 7), compressed, and encrypted with the session key. The resulting data (Figure 8) was then sent to the web server. We also created a user interface that demonstrated how the data could be stored in the database either encrypted or decrypted (Figure 9).

2.4 Source Code

The prototype's source code is organized in two directories in the git repository:

src/html/demo Files for configuration and setup of the demo, as well as the JavaScript implementation.

src/cgi-bin/ Software that runs on the web server in response to the JavaScript running.


Encrypted Survey Demonstration
cats6-12.it.census.gov/demo/survey.c
Search

Survey

This is a secure form.

First name Methuselah

Age 969

Encrypt and Submit!

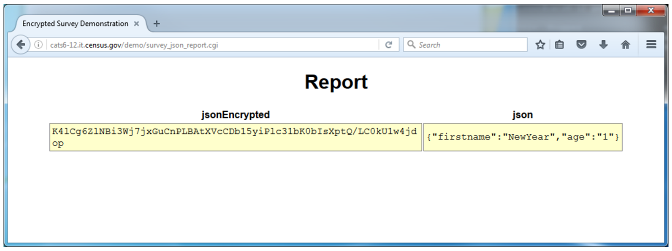
Figure 6. A hypothetical web survey instrument with sample data.

```
{"firstname":"Methuselah","age":"969"}
```

Figure 7. A JavaScript Object Notation (JSON) representation of the web response for Figure 6.

```
K4lCg6ZlNBi3Wj7jxGuCnPLBAtXVcCDb15yiPlc31bK0bIsXptQ/LC0kU1w4jdop
```

Figure 8. The contents of Figure 7, compressed and encrypted.


Encrypted Survey Demonstration
cats6-12.it.census.gov/demo/survey_json_report.cgi
Search

Report

jsonEncrypted	json
K4lCg6ZlNBi3Wj7jxGuCnPLBAtXVcCDb15yiPlc31bK0bIsXptQ/LC0kU1w4jdop	{"firstname":"NewYear","age":"1"}

Figure 9. A user interface that was created for an internal demonstration that shows how either the encrypted JSON values or the JSON structure could be stored in a database.

2.4.1 src/html/demo

This section describes the contents of the src/html/dmeo directory in the git repository.

config.ini This configuration file specifies the location of the database, the database credentials, and a JSON object that specifies the public/private keypair. The key is specified with a JSON data object that specifies the parameters d, e, q, n and p.

createdb.sql The SQL schema for the server database.

dbutil.py A python program that can create the database and list its contents.

demo_random.html A brief HTML demonstration of generating random numbers with JavaScript.

encrypt.js JavaScript routines for using the Web Cryptography API, used for the first prototype.

encrypt_json.js A revised JavaScript encryption system that uses the `window.crypto.subtle.generateKey` function to create an AES-256 key, the `JSON.stringify` method to marshal the form values, and the `encrypt_buff` function to encrypt the JSON string.

jquery-3.1.1.min.js The venerable JQuery JavaScript user interface library.

keyutil.py A utility program or creating and manipulating the RSA keys used in the prototype.

keyutil_test.py Unit tests for the keyutil program.

notes.txt Developer notes.

setup_rel7.sh A bash script that installs the necessary packages on a Red Hat Enterprise License (RHEL) System 7 Linux operating system to run the prototype.

survey_json_report.js JavaScript for displaying the report of successfully encrypted and decrypted survey responses for the second prototype.

survey_report.js JavaScript for displaying the report of successfully encrypted and decrypted survey responses for the first prototype.

2.5 /src/cgi-bin

This section describes the contents of the src/cgi-bin directory in the git repository. This directory contains the Common Gateway Interface files that are used by the prototype; the files are stored in a different directory because the RHEL7 secure configuration that we used required that CGI scripts be stored in a different directory than the HTML and JavaScript files.

report.cgi A CGI script that displays the secure form and shows the public key used to encrypt it.

submit.cgi A CGI script that accepts the RSA-encrypted form, decrypts the parameters, and stores both the encrypted and decrypted values in the database. Note that this script was developed for demonstration purposes: in a production system, *either* the encrypted or decrypted values would be stored in the database, depending on the deployment configuration.

submit_json.cgi A CGI submission script, used by the second prototype.

survey.cgi A CGI script that displays the survey for the first prototype.

survey_json.cgi A CGI script that displays the survey for the second prototype.

survey_json_report.cgi A CGI script that displays a report of the submitted values for the second prototype.

survey_report.cgi A CGI script that displays a report of the submitted values for the first prototype.

3 Conclusion

We developed a system using the World Wide Web Consortium's Web Cryptography API that would provide a second layer of cryptographic protection for survey responses. This second layer was designed to operate even if some kind of TLS decryption or Web Content Filtering system was employed between the respondent and the internet self-response instrument.

Although the system was developed because of concerns over the Department of Homeland Security's EINSTEIN 3A system, the Census Bureau learned during the development of the system that EINSTEIN 3A was designed to monitor *outging* web traffic of U.S. Government employees, and not *incoming* traffic to U.S. Government websites.

Nevertheless, research by academics has determined that a small percentage of internet users have their web traffic monitored using TLS interception techniques similar to those envisioned in the EINSTEIN 3A program. An application-level encryption system would protect respondents against this kind of monitoring. Application-level encryption would also allow sensitive data to be collected by the Census Bureau on internet-connected servers and then to be transferred to internal servers at a later point in time for subsequent decryption. As such, the Census Bureau will continue to develop this technology.

4 Disclaimer

This paper is presented with the hope that its content may be of interest to the general statistical community. The views in these papers are those of the authors, and do not necessarily represent those of the Census Bureau.

References

1. Dick, R. L. Ronald L. Dick, Director, National Infrastructure Protection Center, FBI, Federal Bureau of Investigation, Before the House Energy and Commerce Committee, Oversight and Investigation Subcommittee, Washington, DC, URL https://archives.fbi.gov/archives/news/testimony/issue-of-intrusions-into-government-computer-networks (2001).

2. Rescorla, E. The transport layer security (TLS) protocol version 1.3, URL https://tools.ietf.org/html/rfc8446 (2018).

3. Pennington, R. A. The operational design of the 2020 Census: Overview of the current status. In *JSM 2016—Survey Research Methods Section*, URL http://ww2.amstat.org/sections/srms/proceedings/y2016/files/389667.pdf (2016).

4. Thompsion, J. H. Prepared statement, URL https://www.census.gov/content/dam/Census/about/about-the-bureau/20170215_thompson_testimony.pdf (2017). High-Risk Government Operations Susceptible to Waste, Fraud, and Mismanagement, Before the Committee on Homeland Security and Governmental Affairs U.S. Senate.

5. The 13th biennial Federal Committee on Statistical Methodology (FCSM) policy conference, URL http://www.copafs.org/UserFiles/file/2016FCSMPolicyProgramFinalWebVersion.pdf (2016).

6. Spears, T. Hundreds attend goodbye gathering after StatCan's chief statistician quits. *Ottawa Citizen* URL https://ottawacitizen.com/news/local-news/canadas-chief-statistician-quits-statistic-canada (2016).

7. Department of Homeland Security National Cyber Security Division, United States Computer Emergency Readiness Team(US-CERT). Privacy impact assessment EINSTEIN program: Collecting, analyzing, and sharing computer security information across the federal civilian government, URL https://www.dhs.gov/sites/default/files/publications/privacy_pia_eisntein.pdf (2004).

8. US CERT. *The National Strategy To Secure Cyberspace* (The White House, 2003).

9. EINSTEIN, URL https://www.dhs.gov/einstein (2018). Last access Sept. 2, 2018.

10. Goode, B. & Cantor, J. Privacy impact assessment for EINSTEIN 3—Accelerated (E^3A), URL https://www.dhs.gov/publication/einstein-3-accelerated (2013).

11. Census Bureau. Confidentiality pledge revision notice, URL https://www.federalregister.gov/documents/2016/12/14/2016-30014/confidentiality-pledge-revision-notice (2016).

12. US Census Bureau. Data protection and privacy, URL https://www.census.gov/about/policies/privacy/data_stewardship/federal_law.html (2018). Last Accessed Sept. 4, 2018.

13. Sonic Wall. Deep Packet Inspection of SSL/TLS encrypted traffic (dpi-ssl), URL https://www.sonicwall.com/en-us/products/firewalls/security-services/dpi-ssl (2018). Last accessed Sept. 3, 2018.

14. Palo Alto Networks. Decryption, URL https://www.paloaltonetworks.com/features/decryption (2018). Last accessed Sept. 3, 2018.

15. Desal, S. SSL inspection – issuing CAs and root considerations, URL https://www.globalsign.com/en/blog/what-is-ssl-inspection/ (2017). Last accessed Sept. 3, 2018.

16. O'Neill, M., Ruoti, S., Seamons, K. E. & Zappala, D. TLS proxies: Friend or foe? In Gill, P., Heidemann, J. S., Byers, J. W. & Govindan, R. (eds.) *Proceedings of the 2016 ACM on Internet Measurement Conference, IMC 2016, Santa Monica, CA, USA, November 14-16, 2016*, 551–557, DOI: 10.1145/2987443, URL http://dl.acm.org/citation.cfm?id=2987488 (ACM, 2016).

17. Hans Hoogstraaten (Team leader) and Ronald Prins (CEO) and Daniël Niggebrugge and Danny Heppener and Frank Groenewegen and Janna Wettinck and Kevin Strooy and Pascal Arends and Paul Pols and Robbert Kouprie and Steffen Moorrees and Xander van Pelt and Yun Zheng Hu. Black tulip: Report on the investigation into the diginotar certificate authority breach. Tech. Rep., Fox-IT (2012).

18. Watson, M. Web cryptography API. Tech. Rep. REC-WebCryptoAPI-20170126, World Wide Web Consortium (2017). URL https://www.w3.org/TR/WebCryptoAPI/.

19. Garfinkel, S. L. Notes on meeting of October 13, 2017 (2017). Unpublished.

20. Ozment, A. & Neuman, K. L. Privacy impact assessment for EINSTEIN 3—Accelerated (E^3A), URL https://www.dhs.gov/publication/einstein-3-accelerated (2016).

5 Additional information

5.1 Excerpt from Privacy Impact Assessment for EINSTEIN 3—Accelerated (April 2013)

5.1.1 Relationship Between Participants — Privacy Considerations

"CS&C requires the ability to perform deep packet inspection of known or suspected cyber threats that are identified by EINSTEIN sensors. CS&C screens all data captured by EINSTEIN 1 and EINSTEIN 2 sensors to ensure it is analytically relevant to a known or suspected cyber threat. E^{3}A combines existing analysis of EINSTEIN 1 and EINSTEIN 2 data as well as information provided by cyber mission partners with existing commercial intrusion prevention security services to allow for the near real-time deep packet inspection of federal network traffic to identify and react to known or suspected cyber threats. Network flow records contain only packet header information. Packet inspection tools allow an analyst to look at the content of the threat data, which enables a more comprehensive analysis. Packet Capture may contain information that could be considered PII-like malicious data from or associated with email messages or attachments. CS&C follows SOPs regarding handling of information that could be considered PII including the deletion of any PII unless there is a connection to a known or suspected cyber threat. Packet Capture shows details about the known or suspected cyber threat within the federal network. CS&C analyzes this detailed information and issues warnings, including possible mitigation strategies to the threat.

"In accordance with the SOPs and information handling guidelines, all information that could be considered PII is reviewed prior to inclusion in any analytical product or other form of dissemination, and replaced with a generic label when possible. In some cases, a product may include information that could be considered PII because that information is deemed analytically relevant and necessary to understand the cyber threat. In those instances, the SOPs and information handling guidelines provide for safeguards regarding the marking, dissemination, and handling of the information." [10, p.9]

5.2 Privacy Impact Analysis: Related to Information Sharing

"Privacy Risk: If non-cybersecurity information must be shared outside of DHS, it increases the risk of unauthorized disclosure.

"Mitigation: Information about known or suspected cyber threats collected, analyzed, or otherwise obtained by CS&C may be disclosed for cybersecurity purposes and in furtherance of the DHS cybersecurity mission.

Information collected by E^{3}A or otherwise obtained by CS&C may be disseminated for non-cybersecurity purposes in limited situations when the collected information appears to indicate involvement in activities that may violate laws or otherwise when the sharing is done in the performance of a lawful government function. This may include dissemination for law enforcement/intelligence or administrative purposes unrelated to the protection of an information system from cybersecurity threats, mitigations against such threats, or response to a cyber incident. In such cases, the recipient will be a federal, state, or local law enforcement entity." [10, p.23]

6 Excerpt from PIA Update for EINSTEIN 3—Accelerated (May 2013)

6.1 Reason for the PIA Update — Web Content Filtering

“WCF will provide protection for web traffic5 by blocking access to certain websites that are known to be, or include, malicious content (malware). In addition, WCF will prevent malware from suspicious websites from running on federal civilian Executive Branch D/A systems and networks. Finally, WCF will also detect and/or block phishing attempts as well as the undesirable content that may be included in those attempts.

“WCF categorizes web-based suspicious traffic, to include all URL/URIs and the content of web sessions,6 which allows system operators to specifically allow or disallow certain types of content that is known to be, or includes, malicious content (malware). WCF service can be configured to alert or block on traffic based on the applicable high-confidence cyber threat indicators and commercial signature development technology (used by the ISP) to allow DHS to block and alert against web-based traffic. This will permit traffic suspected by DHS as malicious as well as customer-specific cybersecurity risk protection requirements to alert or block on specific types of traffic. WCF provides this service via a web proxy between the client and the web server it is attempting to access. The proxy will perform the actions such as redirect, prevent, and/or alert on attempted access to certain (i.e., malicious) web content that matches a DHS cyber threat indicator that may look for a specific URL/URI or webpage content.

“WCF capabilities also include in-line Secure Socket Layer (SSL) decryption; malware detection; and advanced analytics. WCF SSL provides visibility into specific types of organizational traffic (including web content) that has been encrypted, for the purpose of protecting that traffic from malicious activity that would otherwise remain hidden by traversing encrypted channels. The capability decrypts web traffic of D/As participating in the E^3A WCF capability for the purpose of detecting and preventing malicious web content on the D/A network. DHS is not interested in the behavior of individuals; decryption is focused on web communications, not communications between individuals. DHS does not use this capability to investigate the behavior or private content of individuals. Malware detection is an inherent part of operating WCF. WCF protects specific federal civilian Executive Branch D/A traffic by using Government-furnished cyber threat indicators to detect malicious activity. Advanced analytics in this context refers to behavior-based (heuristic) threat indicators to identify how a cyber threat or any of the anomalous characteristics of a cyber threat, a computer system, or the data behaves.” [20, pp 2–3].