Electronic Prescription for Controlled Substances: A Cybersecurity Perspective

Samuel Tan, Rebecca Shapiro, Sean W. Smith
Dartmouth College

Abstract

The Electronic Prescription for Controlled Substances (EPCS) is a set of rules published by the Drug Enforcement Administration (DEA) that regulates implementations of electronic prescription systems for controlled substances [1]. EPCS includes requirements two-factor authentication; specifications for electronic prescription applications; and rules governing the signing, transmitting, and receiving of electronic prescriptions [1]. However, this set of regulations overlooks numerous critical aspects of computer security. This paper highlights some key areas in the electronic prescription process outlined by the EPCS regulation that are susceptible to adversarial attacks and provides recommendations for additions to EPCS regulations that would provide greater security for the use of electronic prescriptions.

1. Introduction

On March 31, 2010, the DEA published an Interim Final Rule with Request for Comment entitled "Electronic Prescription for Controlled Substances" (EPCS) in the Federal Register. This set of regulations, which became effective June 1, 2010, aimed to "provide pharmacies, hospitals, and practitioners with the ability to use modern technology for controlled substance prescriptions while maintaining the closed system of controls on controlled substances" [1]. The electronic prescription procedure outlined by EPCS regulations can be summarized as follows:

1) The practitioner writes a prescription using a prescription application that complies with EPCS regulations.
2) The practitioner authenticates to the application using two-factor credentials, issued by an approved Credential Service Provider or Certification Authority (CA) after identity-proofing.
3) After the practitioner has authenticated appropriately to the application, the practitioner signs the prescription with the private key matching his or her digital certificate, or the electronic prescription application signs the electronic prescription. If the prescription was signed against the practitioner's digital certificate, the prescribing application must check this certificate against the certificate revocation list (CRL) of the CA that issued it.
4) The prescribing application archives the prescription, and electronically transmits it to the pharmacy in a manner that aims to ensure that the content and electronic format of the prescription do not undergo any changes.
5) The pharmacy receives the prescription and verifies the digital signature using the pharmacy application, thereby aiming to ensure that transmission integrity was achieved in step 4. If the prescription was signed with the application's cryptographic module, the pharmacy application must also sign the prescription with its cryptographic module to verify its receipt. If the prescription was signed with the practitioner's digital certificate, the pharmacy application must check this certificate against the certificate revocation list (CRL) of the CA that issued it.
6) The pharmacy application archives the prescription, and the pharmacist issues the prescribed controlled substance to the patient.

Unfortunately, these regulations inadequately specify security requirements for many components of this process. In this paper, Section 2 discusses our general security metrics; Section 3 discusses general attack scenarios; Section 4 discusses specific security weaknesses enabled by these scenarios and potential ways to amend the regulations to mitigate these weaknesses; and Section 5 concludes. While our security analysis is fairly straightforward, we believe that the context of its application is significant enough to warrant the attention academics and industry practitioners alike.

2. Security Metrics

We will use the following set of metrics, adopted from the standard security rubrics outlined by Smith and

Marchesini [2], to analyze the security of the electronic prescription procedure specified by EPCS regulations.

Correctness. Only authorized practitioners should be able to issue electronic prescriptions, and controlled substances should only be issued to parties for whom valid prescriptions are intended. We consider this the primary goal.

Integrity. Data stored or transmitted should not be modifiable. Moreover, stored data should not be forgeable. The data we are specifically concerned about are the electronic prescriptions, prescription logs and access controls for the prescribing application, such as CRLs and authentication information. Unauthorized personnel able to modify system access controls and/or electronic prescriptions could illicitly issue or receive electronic prescriptions of controlled substances, which we consider an incorrect outcome. An adversary able to modify, delete or forge electronic prescription logs could hide records of illicit transactions from audits or frame innocent users.

Confidentiality. Data stored or transmitted should not be revealed to parties it was not intended for. The need for confidentiality is two-fold. Firstly, revealing data (i.e. prescriptions or prescription history) could provide an adversary with the information to exploit the system. For example, knowledge of which patients receive prescriptions of a certain drug would provide an adversary with potential targets for theft or extortion. Secondly, confidentiality is necessary for compliance with the *Health Insurance Portability and Accountability Act of 1996* (HIPAA), which requires health care providers protect the privacy of all "individually identifiable health information" in electronic form [3].

Availability. Authorized parties should be able to use the system when they need it. Given the sometimes time-sensitive nature of controlled substance prescriptions, it is crucial to ensure the availability of the systems involved in prescribing, transmitting, and issuing the prescriptions. While EPCS regulations allow for the use of paper prescriptions in cases where the electronic prescription system fails, we think that such a situation is not ideal. Therefore, we treat weaknesses that affect availability as security flaws.

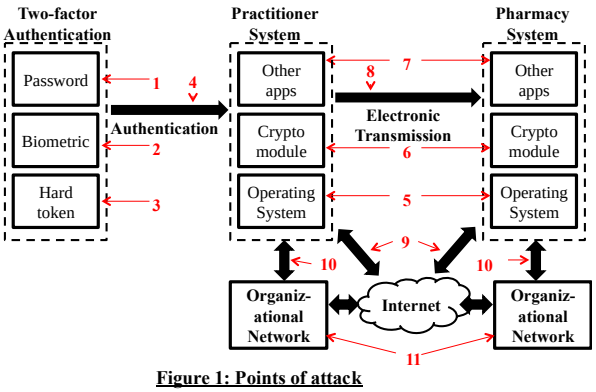

Figure 1: Points of attack

3. Attack Scenarios

The two primary areas of weakness in the electronic prescription procedure mandated by EPCS regulations are 1) the systems (i.e. computers) that the electronic prescription prescribing/processing applications run on (henceforth referred to as the practitioner and pharmacy systems) and 2) the two-factor authentication scheme. In this section, we focus our attention on some representative attack scenarios. We then discuss the security weaknesses that permit these attack scenarios in greater detail in Section 4.

3.1. Practitioner and Pharmacy Systems

A significant portion of the EPCS regulation describes requirements for the electronic prescription application, logical access controls, and creating and signing electronic prescriptions [1]. However, these regulations impose few requirements on the system that the prescribing application will run on, and even fewer on the system that the prescription-processing (pharmacy) application will reside on. The security of these systems and their cryptographic modules are largely determined by Federal Information Processing Standard (FIPS) 140-2 Security Level 1 standards [1], which do not adequately guard against physical or software attacks. The lack of sufficient security specifications on each system's software, configuration, and physical security open them up to several possible attack scenarios, including:

- An adversary subverting the pharmacy or practitioner system (through attack points 5, 7, 9, 10 or 11 in Figure 1) and reconfiguring the system in a manner that prevents it either from issuing or processing prescriptions. This would violate our metric for availability.
- An adversary successfully authenticating to the prescription application, and using it to digitally sign and issue illicit prescriptions. This would violate our metric for correctness.
- An adversary extracting the private key of the cryptographic module responsible for digitally signing prescriptions (through attack point 6 in Figure 1) and using it to digitally sign and issue illicit prescriptions. This would violate our metric for correctness.

3.2. Two-factor authentication

In order to electronically prescribe controlled substances, EPCS regulations require practitioners to obtain

two-factor authentication from a credential service provider [1].

While a two-factor authentication scheme is standard practice in industry and might seem secure, EPCS requirements for each of these three factors (described in Section 4) do not sufficiently guarantee their security. Attack scenarios arising from these requirements include:

- An adversary subverting two of the practitioner's authentication credentials (through attack points 1, 2 or 3 in Figure 1), and using these credentials to authenticate to the prescribing application and digitally sign illicit prescriptions. This would violate our metric for correctness.
- An adversary subverting the practitioner system (through attack points 5, 7, 9, 10 or 11 in Figure 1), launching a man-in-the-middle attack (attack point 4 in Figure 1), collecting valid authentication credentials entered by an authorized practitioner and then using these credentials to authenticate to the prescribing application, and digitally signing illicit prescriptions. This would violate our metric for correctness.
- An adversary eavesdropping on network traffic between the practitioner and pharmacy systems (attack point 8 in Figure 1) and reading the unencrypted electronic prescriptions that have been transmitted. This would violate our metric for confidentiality.

4. Weaknesses

Having outlined several representative attack scenarios, we explore in greater depth the security weaknesses in the electronic prescription process outlined by EPCS regulations that allow such scenarios to arise. For each weakness, we discuss existing preventive measures required by EPCS regulations and then highlight potential threats arising from these current regulations. To keep the discussion from being too vague, we then draw on literature and experience to outline potential attacks against systems which would nonetheless meet the EPCS regulations. However, to make more positive contribution, we then recommend ways the regulations could be extended to mitigate these threats.

4.1. Software Security

We first consider software security issues that EPCS appears to fail to address (corresponding to attack points 5-11 in Figure 1).

Existing measures EPCS regulations impose few requirements on the security and configuration of the software for both the practitioner and pharmacy systems. The regulations only require logical access controls to be established on the use of the practitioner and pharmacy applications, and that "the cryptographic module used to digitally sign data elements must be at least Federal Information Processing Standards (FIPS) 140-2 Security Level 1 validated" [1].

The operating systems that the practitioner and pharmacy's FIPS 140-2 Security Level 1 validated cryptographic signing modules run on must meet additional software security requirements. If the operating system is "modifiable"—that is, its functionality can be modified, added or deleted (e.g. commercially available general purpose operating systems like Windows which software implementations of the cryptographic signing module will run on)—FIPS 140-2 Security Level 1 requires the following:

- The operating system must be *"*restricted to a single operator[6] mode of operation*"* (i.e., concurrent operators are explicitly excluded).
- The cryptographic module must protect its CSPs [Critical Security Parameters] from other processes while it is running.
- Cryptographic software and firmware must be installed so as to prevent their source code and binaries from being view or changed.
- The module's software and firmware shall be protected with an "approved…integrity technique" [4].

If the operating system is not "modifiable" (e.g. firmware embedded in read-only memory of hardware implemented cryptographic modules), FIPS 140-2 Security Level 1 imposes no requirements on it [4].

Threat Existing EPCS regulations are not sufficient to adequately secure the operating system of the practitioner and pharmacy systems. These regulations do not specify user policy configuration on the operating system (access controls are only required to be established for the prescribing application and the cryptographic signing module), do not disallow the running of other processes alongside the cryptographic signing module, and do not mandate that the operating system be kept properly updated. This could lead to a number of attacks, including:

- **Compromise of other services** running alongside the prescribing or pharmacy application on the practitioner or pharmacy system (attack point 7 in Figure 1) which could be used to attack either the system or

[6] "Operator" refers to the user of the cryptographic module

prescribing application directly (for example, see [12]).

- **Compromise of the operating system itself.** Vulnerabilities on an unpatched operating system running on the pharmacy or practitioner system could open either to attack (attack point 5 in Figure 1). Such was the case with the Conficker worm attacks of 2009, which was in part blamed on FDA regulations that prevented the installation of patches from Microsoft that would have blocked the worm from being installed in a timely manner [5].
- **Disabling of critical functions of the host system** by an adversary who has gained access to the administrator account of the operating system (attack point 5 in Figure 1). The adversary could reconfigure the system in a way that either opens it to attacks (e.g. opening ports), or prevents it from carrying out its functions (i.e. issuing and processing prescriptions).

In theory, FIPS 140-2 Security Level 1 requirements should ensure that the prescribing application and its Critical Security Parameters (CSPs)[7] are secure even when running on a general purpose operating system. However, in practice, these measures are not properly tested for. The "single operator mode of operation" requirement can be fulfilled by running a software-implemented cryptographic signing module within a single process of an operating system that uses virtual memory to segregate user process address spaces (e.g. Linux, Windows Server 2008 R2) [6], or by using an operating system that allows only one interactive user to be logged on to the system at a time (e.g. Windows Server 2008 R2) [7]. Since practically all modern general purpose operating systems use virtual memory systems and support single-user login modes, the "single operator mode of operation" requirement is easily fulfilled. However, history has shown us that such operating systems, not properly configured or patched, may be subverted by network, physical, and software attacks (corresponding to attack points 5-7, 9-11 in Figure 1.1).

It is unclear how the second requirement—protecting the module's CSPs from access by other processes—can be thoroughly tested. The derived test requirements are vague about the testing procedure for this requirement, and simply restate the requirement as follows:

> *TE06.05.01….While the cryptographic functions are executing, the same or another tester shall attempt to access … CSPs.* [8]

Given such vague derived test requirements, we are not convinced that the FIPS 140-2 Security Level 1 validated cryptographic signing modules on the practitioner and pharmacy systems can conceal their CSPs from other processes.

Potential attacks We outline attack approaches on the practitioner system described on the website *Metasploit Unleashed* [9]. Depending on the organizational network and system security policy of the institution that the practitioner system belongs to, an adversary could potentially detect vulnerabilities on the prescription system by:

- Using a port scanner like nmap to list open ports on the organizational network, construct a map of services running on this network, and detect which operating systems each machine is running.
- Using a remote security scanner like Nessus to scan all services offered by the practitioner system and determine known security exploits that these services are vulnerable to.
- Using a packet sniffer like Wireshark from another system on the local network to monitor incoming and outgoing packets of the practitioner system.
- Physically accessing the system to obtain system information (e.g. by accessing system information Control Panel > System and Security > Security directory in Windows 7).

Using penetration testing tools like Metasploit, the adversary can utilize information about the practitioner system's operating system and/or vulnerabilities to create an exploit and deliver it as a payload to the system. A successfully deployed payload could subvert the security of the prescription system in numerous ways, including:

- Revealing practitioner passwords through keystroke loggers.
- Allowing for the modification of logical access controls to the prescription application, granting the adversary the ability to create illicit prescriptions.
- Modifying other software running on the operating system in a way that could enable man-in-the-middle attacks[8].

[7] "Critical Security Parameters" is defined in FIPS 140-2 as "security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords and PINs) whose disclosure or modification can compromise the security of a cryptographic module" [4]

[8] Marchesini et al. demonstrated the effectiveness of a "keyjacking" attack, where malicious code injected into Internet Explorer hijacks calls to the CryptoAPI interface, allowing private keys entered by users into the browser to be collected, and used to make actual CryptoAPI calls to generate forged signatures [24]

Clearly, these outcomes violate our metrics for integrity, confidentiality, and correctness. Moreover, if the operating system grants every user administrator privileges, an adversary who is able to log into the pharmacy or practitioner system could uninstall the prescribing application on the practitioner system and/or disable software firewall configurations, thereby preventing the system from performing its function (issuing prescriptions) and/or leaving it vulnerable to network-based attacks. This would violate our metrics for availability and correctness (if network attacks successfully allow an adversary to issue illicit prescriptions).

Possible Mitigation We suggest that an EPCS system also comply with the guidelines in Section 4 of NIST SP 800-44 [10], which outline recommendations for securing a system's operating system, including:

- **Frequent patching and updating of the operating system and applications** to prevent adversaries from exploiting any known vulnerabilities.
- **Uninstalling or disabling unnecessary services and applications** to eliminate opportunities for adversaries to attack the practitioner or pharmacy system.
- **Configuring user access to the operating system** to ensure that only authorized users can make changes to the operating system.
- **Configuring access privileges for system resources** to ensure that unauthorized users cannot modify or view prescription information or logs, thereby ensuring their confidentiality and integrity.
- **Installing additional safeguards** such as firewalls, anti-virus, and malware detection programs to further secure the system against attacks.
- **Conducting security audits on the operating system** in the form of vulnerability scanning or penetration testing to ensure that pre-existing security measures are adequate and to detect existing vulnerabilities.

While these NIST recommendations were written for securing operating systems running on web servers, we posit that they are equally relevant to practitioner and pharmacy systems covered by EPCS regulations.

4.2. Network Security

We next consider network security issues that EPCS appears to fail to address. (These correspond to attack points 9-11 in Figure 1.)

Existing measures EPCS regulations do not specify any network security requirements for the practitioner and pharmacy system or on the network to which either system is connected.

Threat The practitioner and pharmacy systems may be connected to the Internet to communicate with each other, if not to their own internal organizational networks. Given the lack of any network security requirements, the practitioner and pharmacy systems will be open to attacks via the networks they are connected to (attack points 9 and 10 in Figure 1), and attacks on these networks themselves (attack point 11). Failure to properly enforce network security has already compromised the security of numerous institutions, medical and otherwise.

Potential attacks An attack *via* the network leverages insecure system network configuration (usually open ports or network-facing services) to detect and exploit weakness in system software. A textbook example of such an attack was described in Section 3.1. The lack of any network security requirements in EPCS allow the practitioner or pharmacy system's to be open to these network attacks while still being in compliance with these regulations.

Alternatively, attacks *on* the network subvert network infrastructure/protocols to reroute network traffic in a malicious way. An example of such an attack is Address Resolution Protocol (ARP) cache poisoning, where a malicious machine sends false ARP replies to other machines on the network in order to trick those other machines into thinking that the malicious machine is in fact a different, trusted machine [2]. Attacks on the Domain Name System (DNS), such as DNS cache poisoning, DNS spoofing, and DNS ID hacking, cause client machines to receive incorrect responses to their DNS queries, which then redirects their network traffic to an attacker-controlled machine [11]. If the attacker can forge a DNS response, the attacker can then masquerade as the service that is being queried by the client and perform a man-in-the-middle attack.

Possible Mitigation We suggest that EPCS systems also comply with the guidelines in Section 8 of NIST SP 800-44: Guidelines on Securing Web Servers, including:

- **Proper organizational network layout** to ensure that the practitioner and pharmacy systems are properly protected by the organization's network security elements, and that these systems are not unduly exposed to threats from insecure systems within the organization's internal network.

- **Proper firewall configuration** to block all inbound traffic to the practitioner and pharmacy system other than those necessary to carry out prescription functions, and to log potential intrusion attempts.
- **Intrusion detection and prevention systems** to prevent and notify administrators of intrusion attempts.
- **Network switches** to guard against eavesdropping and thus ensure confidentiality of prescription information [12].

Again, while these recommendations were written for securing web servers, they are equally relevant to securing the network infrastructure of practitioner and pharmacy systems covered by EPCS, which are similarly exposed to either internal networks or the Internet.

4.3. Physical Security

We now consider physical security issues (attack points 3 and 6 in Figure 1).

Existing measures Three modules are crucial to ensuring the correctness of the electronic prescription procedure mandated by EPCS regulations: the cryptographic signing modules used to sign outgoing and incoming prescriptions on the practitioner and pharmacy systems respectively, and the hard token used by the practitioner for two-factor authentication.

EPCS regulations require that cryptographic signing modules used by the prescribing and pharmacy applications adhere to FIPS 140-2 Security Level 1 requirements [1]. However, FIPS 140-2 Security Level 1 imposes weak requirements on the physical security of these modules, namely that the cryptographic signing module is made out of "production grade equipment", and that if a maintenance/debugging mode exists, "all plaintext secret and private keys and CSPs [Critical Security Parameters] shall be zeroized when the maintenance access interface is accessed" [4].

Similar regulations apply to the hard token, should the practitioner choose to use one for two-factor authentication. EPCS regulations state that the hard token "must be separate from the computer to which it is gaining access and must meet at least the criteria of FIPS 140-2 Security Level 1" [1]. As stated above, FIPS 140-2 Security Level 1 requires only that the token is made out of "production grade equipment" and is designed to zeroize keys during maintenance access [4].

If the cryptographic signing module of the practitioner or pharmacy system is fully implemented in software, physical security requirements do not apply [4]. However, EPCS regulations require that "when the signing module is deactivated, the [prescribing] application must clear the plain text password from the application memory to prevent the unauthorized access to, or use of, the private key" [1].

Threat Given the lack of physical security requirements for standalone hardware cryptographic signing modules and hard tokens, an adversary with physical access to any of these devices could easily perform a range of physical hardware attacks on them, including:

- **Attacks on the maintenance access interface**, which aim to disable the mechanism that detects maintenance mode access and triggers the zeroizing of keys.
- **Power analysis attacks**, which exploit the information leaked by the token's hardware (i.e. power consumption) to uncover underlying cryptographic secrets. Kocher et al. note that these attacks are "easy to implement, have a very low cost per device, and are non-invasive, making them difficult to detect" [13]. Such attacks are mentioned in Section 4.11 of FIPS 140-2, but FIPS 140-2 certified cryptographic modules are not required to prevent such attacks [4].

If the cryptographic signing module is implemented in software, the system that it operates on is susceptible to all the above-mentioned attacks as well, since FIPS 140-2 does not state any physical security requirements on the system the module runs on [4]. Software modules are also open to other side-channel attacks on the host system, such as cache timing attacks, which observe timing patterns introduced by the cache (from cache hits and misses) to deduce the state of a cryptographic algorithm. Percival demonstrates such an attack on the Pentium 4 Processor to effectively extract keys for OpenSSL [14], which is FIPS 140-2 Security Level 1 validated [6].

Moreover, if cryptographic keys are stored in memory, it is unlikely that purging them entirely from memory would be easily achievable. Chow et. al found that many commercially available applications take few measures to limit the "lifetime" (i.e. presence in memory) of sensitive data (e.g. passwords), transmitting these data across memory without any provisions to clear them [15]. Memory attacks could also still be performed while the system is live.

All of the abovementioned attacks would compromise CSPs, potentially subverting the hard token as an authentication factor, or allowing unauthorized use of the prescribing application to issue illicit prescriptions. Both outcomes would violate our metric for correctness.

Potential attacks The maintenance access interface could be attacked using simple tools. Suppose the internals of the cryptographic module can be accessed via a removable panel, and a switch on the inner side of this panel detects when it is opened, and zeroizes CSPs in response. Simply drilling a hole through this panel and applying glue on this switch could disable it and allow the cryptographic module to be accessed through the panel with the CSPs intact.

Performing a power analysis attack on a hard token or standalone hardware cryptographic signing module requires only power measuring equipment and knowledge of the encryption algorithm used by the hard token. The former is cheaply and easily obtainable, and the latter is usually publicly available information. An adversary would merely have to physically connect the power measuring device, typically a simple resistor, in series with the power or ground input of the device, and calculate the current by taking the difference in voltage readings across the resistor divided by its resistance [13]. The adversary could then produce a graph of the current readings across the duration of a cryptographic operation (known as a *trace*) and either analyze it directly (also known as *Simple Power Analysis (SPA)*), or use statistical functions and error-correction techniques (also known as *Differential Power Analysis (DPA)*) to reveal private keys [13]. Örs et al. describe an implementation of a power analysis attack on an Application-Specific Integrated Circuit (ASIC) Advanced Encryption Standard (AES) implementation [16].

Memory attacks could also extract the private keys of software implementations of the cryptographic signing module while the prescription application is still running. Halderman et al. demonstrate examples of "cold boot attacks" that exploit the remanence of data in DRAM to extract cryptographic secrets from memory images obtained from live systems [17]. The attack involves cooling memory chips with a refrigerant, cutting power to the system, transferring the memory chips to another computer before the data within decays, and reading this data to recover private keys.

Possible Mitigation We suggest a FIPS 140-2 Security Level 3 validation requirement for the hard token, which requires:

- **Tamper-evident seals or coatings** are placed on cryptographic signing modules to expose attempts to physically access plaintext cryptographic keys and CSPs in the module. This will more strongly ensure that attempts to physically attack the system will be exposed during a later audit or investigation.
- **Physical security mechanisms** that have a "high probability of detecting and responding to attempts at physical access, use or modification of the cryptographic module", such as "strong enclosures" and "tamper detection/response circuitry that zeroizes all plaintext CSPs when the removable covers/doors of the cryptographic module are opened". This will render the success of physical attacks on the system's cryptographic signing module more unlikely.
- **Trusted paths** for entry or output of plaintext CSPs, using "ports that are physically separated from other ports", or "interfaces that are logically separated using a trusted path from other interfaces" [4].

For software implementations of the cryptographic signing modules, we recommend imposing the same FIPS 140-2 Security Level 3 physical security requirements to secure the system that the signing module operates on. We acknowledge that this level of validation comes at a higher cost, but do not believe that lower FIPS 140-2 Security Levels are sufficient to achieve our goal of correctness.

To decrease the likeliness of success of power analysis attacks, we propose a requirement that approved hard tokens and standalone hardware cryptographic signing modules implement and document at least one hardware or software power analysis countermeasure. Broad categories of such counter measures are outlined in Section 5 of [18]. To increase the difficulty of performing memory attacks, particularly the "cold boot attack" described above, we recommend that private keys be cleared from memory at fixed, short intervals, or as soon as they are no longer needed. Further countermeasures outlined in [17] could also be implemented.

4.4. Password

We now consider password security issues (attack points 1 and 9 in Figure 1).

Existing Measures Section 1311.102 of EPCS regulations state that practitioners "must not share the password or other knowledge factor, or biometric information, with any other person", and that the practitioner "must not allow any other person to use the token or enter the knowledge factor or other identification means

to sign prescriptions for controlled substances". No requirements for the password policy are specified [1].

Threat Given the lack of requirements on the password policy used in the two-factor authentication scheme, the following issues could arise that subvert the password as an authentication factor:

- **Writing down passwords** in visible locations close to the system these passwords authenticate to, which grants a physically proximate adversary easy access to the system.
- **Reusing of passwords on other websites**, which indirectly exposes the two-factor authentication scheme to the risk of those other websites being subverted by an adversary.
- **Weak knowledge factors for password retrieval** (e.g. country of birth, mother's maiden name), which, with the help of a search engine and social networks, could potentially allow an adversary to retrieve a practitioner's password.
- **Passwords stored in plaintext** on online database, which could be seized by an adversary who is able to gain access to them[9].
- **Weak hashing algorithms** applied on passwords that can be brute-forced by attackers who have access to these hashed passwords using a table of precomputed hash values (a.k.a. rainbow tables).
- **Phishing and social engineering attacks** that trick a practitioner into giving away his or her password.

Potential attacks Attacks that aim to obtain passwords correspond to attack point 1 in Figure 1. An adversary who is able to gain physical access to the area around the practitioner or pharmacy machine could potentially obtain passwords to authenticate to either system from written copies (e.g. post-its) posted nearby. A short password, potentially the result of a maximum character length-requirement policy, could be guessed easily using a traditional brute force or dictionary attack. Any of these simple attacks, if successful, could allow an adversary with access to another authentication factor to falsely authenticate to the prescribing application and issue illicit prescriptions. This would violate our metric for correctness.

Moreover, password reuse dramatically broadens the attack surface for an adversary, essentially opening the practitioner system to attacks from other Internet-facing services (attack point 9 in Figure 1). Even if the practitioner passwords are sufficiently protected, an adversary might still be able to obtain them by subverting weaker websites or services on which these passwords are reused.

Possible Mitigation We recommend that EPCS regulations require implementation of some or all of the password management best practices outlined in NIST SP 800-118, Guide to Enterprise Password Management [19], particularly those outlined in Section 3, in order to make it more difficult for an adversary to obtain practitioner passwords and gain unauthorized access to the prescription application.

4.5. Biometrics

We now consider biometric security issues (attack point 2 in Figure 1).

Existing measures Section 1311.116 of EPCS regulations lists security requirements for biometrics used in the two-factor authentication. Included in these regulations are specifications for the performance of the biometric subsystem and matching software, and requirements for integration or physical proximity of the biometric subsystem to the practitioner system [1]. Additionally:

- The biometric subsystem "must protect the biometric data (raw data or templates), match results, and/or non-match results when authentication is not local" and biometric data is "sent over an open network" [1].
- The biometric system "must conform to Personal Identity Verification authentication biometric acquisition specifications, pursuant to NIST SP 800–76–1" [1].

Since NIST SP 800-76-1 only specifies biometric acquisition methods for fingerprint and facial image biometrics [20], we assume that these are the only two forms of biometrics approved for use under EPCS.

Threat The fact that EPCS regulations do not require that "local" biometric data or match results be protected exposes the biometric subsystem to attack. The term "local" is vague, but likely refers to transmission that does not take place over an open network (i.e. between the different, physically integrated/proximate components of the biometric subsystem). An adversary with physical access to the biometric subsystem could, through physical means, read or intercept unencrypted data transmitted between components of the biometric subsystem, thereby allowing the adversary to launch indirect attacks. Ratha et al. outline examples of such

[9] Two such attacks in 2012 led to 100,000 and 450,000 plaintext passwords being extracted from the Institute of Electrical and Electronics Engineers (IEEE) [26] and Yahoo [25] respectively.

attacks, including replaying biometric data to the feature extractor and overriding matcher decisions or the matcher itself [21].

Potential attacks Attacks that aim to subvert the biometric authentication subsystem correspond to attack point 2 in Figure 1. Galbally et al. describe indirect attacks on fingerprint biometric systems using the "hill climbing technique" which "uses the score given by the matcher to iteratively change a synthetically created template until the score exceeds a fixed decision threshold and access to the system is granted", citing several documented examples of such attacks [22]. Adler demonstrates an attack using a similar technique on facial recognition systems, where a facial image is repeatedly modified based on match score values until a match score with a high probability of matching the desired face is attained [23]. These attacks are very real threats as they require only knowledge of the format of images presented to the feature extractor and access to the score output from the matcher. Successfully subverting the biometric subsystem using such an attack could allow an adversary with access to another authentication factor to falsely authenticate to the prescribing application and issue illicit prescriptions. This would violate our metric for correctness.

Possible Mitigation To prevent indirect attacks on the biometric subsystem, we recommend:

- **Encrypting and authenticating data transmitted locally between different components of the biometric subsystem.** This would prevent an adversary from intercepting, reading or modifying any data flowing through the biometric subsystem. The output of the matcher should also be encrypted to prevent "hill climbing" attacks that rely on being able to read this output.
- **Physically securing the biometric subsystem.** Protecting the biometric subsystem with tamper-resistant or tamper-responding installations would prevent an adversary from physically accessing its internals and intercepting or modifying data transmitted or stored within the system. A FIPS 140-2 Security Level 3 validation requirement would ensure that the biometric subsystem has physical security mechanisms in place to detect and respond to physical tampering [4]. As before, we acknowledge that this level of validation comes at a higher cost, but do not believe that lower FIPS 140-2 Security Levels will adequately defend the biometric subsystem against physical attacks.
- **Implementing liveness detection methods.** These methods detect synthetically generated biometrics using measures of "liveness" (i.e. how likely the biometric is to belong to a live human), thereby guarding against direct attacks and "hill climbing" attacks. Galbally et al. describe examples of such countermeasures [22].

4.6. Transmission

Finally, we consider transmission security issues (attack point 8 in Figure 1). Due to space constraints, we treat this more concisely; our full technical report will contain more details.

EPCS regulations require that electronic prescriptions are protected from modification during transmission. While this ensures the integrity and thus correctness of electronic prescriptions, there are no measures in place to prevent eavesdropping by third parties or to guarantee the delivery of the electronic prescriptions. In particular, EPCS does not require that the transmitted prescriptions are encrypted, which goes against standard practice. An adversary, using a packet sniffer like Wireshark on another machine on the practitioner/pharmacy system's network, could intercept and read unencrypted outgoing electronic prescriptions. An adversary in control of a botnet could potentially launch a DoS attack on the pharmacy system by flooding it with invalid prescriptions. The pharmacy system would be forced to devote its system and network resources to authenticating and rejecting these prescriptions.

Possible Mitigation We recommend using the TLS protocol to establish a secure cryptographic tunnel between the practitioner and pharmacy system through which the electronic prescriptions will be transmitted.

5. Conclusion

EPCS regulations, in their current form, do not adequately protect the electronic prescription process from adversarial attacks. However, many of these loopholes—particularly those related to software, network and password security—are addressable by amending EPCS regulations to require basic best practices. Other attacks that are harder to defend against, such as physical attacks on the hard token or practitioner or pharmacy systems, should be combated by imposing more stringent security requirements that significant raise the difficulty and cost of performing such attacks to potential adversaries. While these countermeasures may increase costs, we believe that these are necessary costs to achieve the security goals outlined in Section 2.

Acknowledgements

This work is supported in part by the US National Science Foundation's Trustworthy Computing award #0910842 and is part of the Trustworthy Information Systems for Healthcare project sponsored by Dartmouth's Institute for Security, Technology, and Society; however, views and conclusions are the authors' alone.

References

[1] Drug Enforcement Administration, "Electronic Prescriptions for Controlled Substances (EPCS)," 31 March 2012. [Online]. Available: http://www.deadiversion.usdoj.gov/ecomm/e_rx/.

[2] S. Smith and J. Marchesini, The Craft of System Security, Addison-Wesley Professional, 2007.

[3] United States Department of Health and Human Services, Summary of the HIPAA Privacy Rule, Office for Civil Rights.

[4] National Institute of Standards and Technology (NIST), Federal Information Processing Standards (FIPS) 140-2: Security Requirements for Crytographic Modules, 2 ed., vol. 140, National Institute of Standards and Technology (NIST), 2001.

[5] E. Ackerman, "Conficker worm hits hospital devices," Mercury News, 1 May 2009. [Online]. Available: http://www.mercurynews.com/breakingnews/ci_12257206.

[6] OpenSSL Team, "OpenSSL FIPS 140-2 Security Policy," Open Source Software Institute, 2012.

[7] Microsoft Corporation, "Windows Server 2008 R2 BitLocker™ Drive Encryption Security Policy For FIPS 140-2 Validation," Microsoft Corporation, 2009.

[8] Computer Security Division, Information Technology Laboratory, NIST, Derived Test Requirements for FIPS PUB 140-2, National Institute of Standards and Technology (NIST), 2011.

[9] Offensive Security Ltd., "Metasploit Unleashed," [Online]. Available: http://www.offensive-security.com/metasploit-unleashed/Main_Page.

[10] M. Tracy, W. Jansen, K. Scarfone and T. Winograd, Guidelines on Securing Web Servers, 2 ed., vol. 800, National Institute of Standards and Technology (NIST), 2007.

[11] F. Carli, "Security Issues with DNS," SANS Institute, 2003.

[12] J. H. Allen, K.-P. Kossakowski, G. Ford, S. Konda and D. Simmel, "Securing Network Servers (2000)," CMU/SEI-2000-SIM-010, April 2000.

[13] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," in CRYPTO '99 Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, 1999.

[14] C. Percival, "Cache missing for fun and profit," in Proceedings of BSDCan 2005, 2005.

[15] J. Chow, B. Pfaff, T. Garfinkel, K. Christopher and M. Rosenblum, "Understanding data lifetime via whole system simulation," in Proceedings of the 13th conference on USENIX Security Symposium, Berkeley, 2004.

[16] S. B. Örs, F. Gürkaynak, E. Oswald and B. Preneel, "Power-Analysis Attack on an ASIC AES implementation," in International Conference on Information Technology: Coding and Computing (ITCC'04), Las Vegas, 2004.

[17] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum and E. W. Felten, "Lest We Remember: Cold Boot Attacks on Encryption Keys," in Proc. 2008 USENIX Security Symposium, San Jose, 2008.

[18] M. Joye and F. Olivier, "Side-channel analysis," Encyclopedia of Cryptography and Security, pp. 571-576, 2005.

[19] K. Scarfone and M. Souppaya, Guide to Enterprise Password Management (Draft), vol. 800, National Institute of Standards and Technology (NIST), 2009.

[20] C. Wilson, P. Grother and R. Chandramouli, Biometric Data Specification for Personal Identity Verification, National Institute of Standards and Technology, 2007.

[21] N. K. Ratha, J. H. Connell and R. M. Bolle, "An Analysis of Minutiae Matching Strength," in AVBPA '01 Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication, 2001.

[22] J. Galbally, J. Fierrez and J. Ortega-garcia, "Vulnerabilities in Biometric Systems: Attacks and Recent Advances in Liveness Detection," DATABASE, vol. 1, no. 3, 2007.

[23] A. Adler, "Sample images can be independently restored from face recognition templates," in Canadian Conference on Electrical and Computer Engineering, 2003. IEEE CCECE 2003, 2003.

[24] J. Marchesini, S. W. Smith and M. Zhao, "Keyjacking: Risks of the Current Client-side Infrastructure," in 2nd Annual PKI Resarch Workshop, NIST, 2003.

[25] P. Wagenseil, "450,000 Yahoo! Passwords Stolen in Data Breach," SecurityNewsDaily, 7 December 2012. [Online].

[26] O. Williams, "IEEE data breach: 100K passwords leak in plain text [Update]," Neowin, 26 September 2012. [Online].