LESSON 8: FROM KEY DECISIONS TO CROMERR-COMPLIANT SOLUTIONS

FOUR CRITICAL ROADMAP ITEMS

Lesson 8 focuses on four CROMERR System Checklist items that are closely related to the two key decisions discussed in Lesson 7, namely:

- Item 3: Issuance (or Registration) of a Signing Credential in a Way that Protects it from Compromise;
- Item 5: Binding of Signatures to Document Content;
- Item 13: Credential Validation; and
- Item 18: Creation of COR.

These four Roadmap items are especially important to assuring both that:

- CORs maintained by your system truly represent what was submitted; and
- Any associated electronic signatures can be proved to be authentic.

For each of these items, Lesson 8 provides both:

- General advice on how they can be addressed; and
- Specific examples of successful solutions drawn from already approved applications.

ITEM #3: ISSUANCE OF A SIGNING CREDENTIAL

As described in Lesson 7, Decision 1—the type of credential used—affects what is required to ensure that the credential is protected from compromise, as required by checklist Item 3. Item 3 is critical to proving the authenticity of the signatures it is used to create—if the credential is compromised, then there would be no way to control who may use it to create associated signatures.

Generally, the solution for Item 3 needs to include answers to the following questions:

HOW IS THE CREDENTIAL ISSUANCE PROCESS LINKED TO IDENTITY PROOFING (ITEM 1)?

If there is any uncertainty about who the credential was actually issued to (or registered for), then there is no way to tell who has it. Some approved systems send an email to the email address provided by the user on the Subscriber Agreement submitted to satisfy identify-proofing requirements.

In one approach to credential issuance, the link is provided by a verification key generated by the system and sent to the email address that the user provided on the Subscriber Agreement (which was submitted to satisfy identity-proofing requirements). The verification key is supplemented by requiring the user to enter the answer to a preset security question.

In another approach, the link is provided by a hyperlink generated by the system and sent to the email address that the user has provided on the Subscriber Agreement (which was submitted to satisfy

identity-proofing requirements). The hyperlink is supplemented by requiring the user to enter a password and provide the answers to two preset security questions.

What kind of credential is it?

For example, is it a PIN or a password combined with the answer to a challenge question? Is it PKI certificate associated with a private–public key pair?

What is the actual process for issuing or registering the credential?

What are the actual steps a user takes to receive or register his or her credential? Does the user log into the system or enter a password?

One process could involve having each user log on to the system with a verification key received via email, answer a security question, and create a password subject to password-strength requirements.

Another process could be to have the user log on to the system with the hyperlink received via email, provide his or her password, answer two security questions, and download the certificate package created by the PKI certificate authority.

How is the credential protected from compromise as it is issued or registered?

That is, what kind of security is provided for the transaction (e.g., is SSL or TLS used)?

One approach could be to use a password creation session protected with SSL or TLS. An alternative approach could be to encrypt the private key and secure the download session with SSL.

How is the credential protected from compromise or tampering as it is stored in your system?

That is, what kind of security is there, and does it include some kind of encryption of the credentials?

Some systems use passwords and security question answers that are one-way hashed, and stored in the system in that form. Other systems use a private key that is encrypted and stored only on the user's workstation. The private key may only be decrypted with a password available only to the user.

Is there a process to allow the user to change his or her credential?

And if so, how does your system ensure that only the legitimate account holder is able to do this?

One process could require users who wish to change their password to enter the account's current password and answer a security question. Alternatively, in cases where the credential or password is lost or compromised, the user could be required to re-register and apply for a new credential.

Item #5: Binding Signature to Document Content

Both key decisions—type of credential and definition of COR—affect Item 5.

Recall that Decision 1—type of credential—affects signature binding in terms of the added functionality provided by PKI-based digital signatures. Decision 2—definition of COR—also affects this item. The COR will have to include both the locked document and the mechanism that provides the lock. Locking a document to be maintained as a discrete file will be very different than locking data elements in a database or on a paper printout.

Item 5 is critical to proving that the COR reflects what was signed and submitted. If the COR can be changed without detection after signature and submission, then there is no way to confirm the actual submission or to what the signer attested.

Generally, the solution for Item 5 needs to include answers to the following questions:

What are the steps in the signature process?

For example, when does the signer provide the credential that executes the signature? What happens before and after? Which steps occur online or offline?

Signing could require an offline digital signature executed for the file containing the submission and an online entry of a password in conjunction with a review of the Certification Statement.

Alternatively, the process could include having a user log into the system by entering his or her password and answering a challenge question. The user would then be presented with the opportunity to review the document being signed and the certification statement. To sign, the user would then enter his or her password again and press a Submit button.

What constitutes the actual signature?

For example, is it the PIN and answer to a challenge question provided by the signer? Or is it the digital signature created with the signer's private key?

Like the Signature Process, the signature could have two parts: a digital signature executed offline and a password entered by the user in conjunction with viewing the Certification Statement.

Or, the signature could simply be the password entered by the user.

At what point in the Submission Process is the document actually locked, and what is the locking mechanism?

For example, is the document locked by the execution of the signer's digital signature? Or is this a hashing function (or digital signature) executed by your system once the submission reaches your server?

For example, execution of the digital signature could lock the document. It would be created by calculating the hash value of the content being signed and then encrypting the hash with the user's private key.

Or, upon receiving the submission, the system could calculate a hash value for the submitted data file.

HOW ARE THE LOCKED DOCUMENT AND THE LOCK (E.G., THE HASH VALUE) INCORPORATED INTO THE COR?
For example, are these components of the COR?

The COR could include the document content (the locked document) together with its digital signature (the lock).

The COR could also include the submitted data file (the locked document) and the hash value (the lock) of that file.

HOW IS THE LOCK PROTECTED FROM TAMPERING?
For example, if the lock is a hash value, what would prevent someone with back-end access to your system from changing the COR, recalculating the associated hash, and substituting these for the originals?

The lock could be the user's digital signature, which is the hash of the document content encrypted with the user's private key. Someone who wishes to hide a change in the document content by replacing the lock with a new one would have to access the user's private key to execute a new digital signature with it. So, the security of the user's private key protects the lock from tampering.

Alternatively, the hash value could be protected from tampering by tightly controlled system access, redundant storage, and providing it back to the signer or submitter. Someone who wishes to replace this hash value with a recalculated version—to hide changes in the COR—would have to defeat system access controls and access all the copies of the original, including the one in the signer's or submitter's custody.

ITEM #13: CREDENTIAL VALIDATION

Decision 1—the type of credential used—affects what is required to validate it. Item 13 is critical to proving the authenticity of the signatures it is used to create. If the credential is not valid, then it may not belong to the signer identified in the submittal, or it may be compromised. In either case, there would be no way to prove that the identified signer is the individual who actually signed the submittal.

Generally, the solution for Item 13 needs to include answers to the following questions:

HOW DOES THE SYSTEM DETERMINE THAT THE CREDENTIAL IS GENUINE—THAT IT WAS ACTUALLY ISSUED AS A PART OF THE REGISTRATION PROCESS?

In the case of PINs and passwords, this may simply be a matter of looking up the credential in a table maintained by the system. In the case of a third-party credential, such as a PKI certification issued by a certificate authority, this may require interaction with the third-party.

For example, a system could verify that the certificate presented by the user was issued by the organization and has not been placed on any certificate revocation list (CRL).

Or, a system could compare the hashed version of a password entered by the user with the hashed version the system stores with the user's account information to confirm that they match.

How does the system determine that the credential actually belongs to the signer identified in the submittal?

For PINs and passwords, this may be a simple table look-up function, but for third-party credentials, this may require interaction with that party to verify identifying information embedded in the credential itself.

For example, a system could confirm that the identified signer is the individual identified by the certificate, which associates that person with a public key.

Or, similar to the way a system determines a credential is genuine, a system could compare and match the hashed version of a password entered with the hashed version of the password stored in the user's account.

How does the system determine that the credential was not compromised at the time of signature?

Addressing this issue normally requires the validation of a second-factor (known as second-factor authentication), which is some item uniquely within the control of the identified signer that can be used to prove that it was this individual—and no one else—who presented the credential to sign the submittal.

One approach to do this is to have the public key decrypt the digital signature, and thus confirm that it was executed with the associated private key. The private key would be protected by a password. To confirm that the password has remained within the exclusive control of the identified user, he or she would be required to answer a challenge question at log-in, which provides a second authenticating factor.

Or, a system could rely on a challenge question as a second authenticating factor in conjunction with a PIN- or password-based credential.

The most commonly used second-factor is the answer(s) to one or more pre-set challenge questions—although there are also more technologically sophisticated options available.

Item #18: Creation of COR

Both key decisions affect Item 18. Recall how Decision 2—definition of COR—affects the process of creating the COR and showing that it represents what was submitted. Decision 1—type of credential—

also affects Item 18. By helping to determine the Signature-Binding Process (Item 5), the choice of type of credential also affects how the COR can be shown to be a "true and correct" copy of the submittal.

Item 18 is closely connected with Item 5, and many successful applications address both items together under Item 5. In any case, both items are critical to proving that the COR reflects what was signed and submitted.

Generally, the solution for Item 18 needs to include answers to the following questions:

What constitutes the COR for your system?

If you have not addressed this question under earlier items (for example, under Item 5), Item 18 is the place to list, in detail, the components of the COR and how they are packaged together.

For example, the COR could include the submitted data, date and time of receipt, associated electronic signatures, and metadata to document the COR's integrity.

How does the COR provide a "true and correct" copy of the submittal—"true and correct" refers to having the same informational content (but not necessarily in the same format)?

The solution to this question may be closely related to the solution for Item 5, since, for example, the secured hash value that binds the signature to the document content may also help show that the COR containing this content is true and correct. A solution will also need to explain how the COR and any associated metadata (such as the hash value) are secured from tampering or destruction.

The contents of the COR could be digitally signed with a system certificate as soon as the submission is received, with both the digital signature and associated key secured by the system.

Or, the submission could be digitally signed at the user's workstation with a temporary private key that is not recoverable once the user session concludes. Decrypting the signature (with the associated public key stored with the COR) and comparing it with a recalculated hash of the signed document could assure the COR's integrity. Since the private key is not recoverable, no counterfeit signature could be generated to hide unauthorized changes to the COR.

How does the COR include any associated e-signatures, and how does their inclusion avoid credential compromise?

CROMERR requires that the COR include any associated electronic signatures. When the signature includes the entry of a PIN or password, they are often included in an encrypted or hashed form to avoid compromising the PIN- or password-based credential.

For example, the COR could include e-signatures as hashed or encrypted passwords.

How does the COR preserve evidence of how it appeared to the signer when presented in a human-readable format?

If the COR is simply a PDF in a human-readable format, then this fact provides the answer. Otherwise, CORs for electronically signed submittals will need to include the formatting mechanism.

For example, if the COR is maintained as an XML file, then the COR should include the XSL style sheet used in conjunction with the file to present it back to the signer.

Additional Sample Solutions

The U.S. EPA CROMERR website includes a number of state-developed and approved CROMERR applications that are available for reference.

These sample applications include resources such as completed CROMERR System Checklists, completed Cover Sheets (Pages 1 and 2), and success stories of the related effort.

Another resource provided by EPA to help applicants is the CROMERR Application Challenges and Solutions document (first introduced in Lesson 3 of this training), which is available on the Tools for States, Tribes, and Local Governments page of the CROMERR site. This document describes common challenges identified in CROMERR applications but also provides examples of solutions used by CROMERR-approved systems.

All of these resources are provided by EPA to help applicants identify ways to meet CROMERR requirements.

Helpful Resources

Below is a list of the various tools and resources referenced in this training, which are also available through the CROMERR website.

Location	Description
Cover Sheet Page 1 and Page 2	This two-page application cover sheet template provides a format for capturing specific and necessary information for the application.
CROMERR System Checklist Template	This 13-page template provides a format for capturing information regarding the system and how it meets specific CROMERR requirements.
Sample Successful Application Checklists	These are examples of EPA-approved application checklists that describe how the electronic document receiving system meets the applicable § 3.2000 CROMERR requirements and the application documents that provide an overview of the systems and the electronic submissions received by the

Location	Description
	systems.
CROMERR Application Challenges and Solutions	This document presents common challenges identified in CROMERR applications received by EPA. For each challenge, it presents the CROMERR issue or deficiency, examples of effective approaches to resolving them, and the EPA-approved systems that use these effective approaches. This document should be used by state and local environmental agency officials and system managers to help them in planning systems that are CROMERR-compliant, preparing CROMERR applications, and responding to notices from EPA of issues and deficiencies for submitted applications.
Challenge Question Second Factor Approach	This document describes the requirements and potential approaches to using a second factor. EPA has determined that to meet the CROMERR requirements for priority reports, a system using PIN or password must be accompanied by some other identifier that together with the PIN or password will be sufficient to prove that the e-signature has not been compromised. One approach is to use the PIN or password in conjunction with a challenge question to create an e-signature.
Fact Sheets	These files contain information about how CROMERR affects agencies, submission reports, and EPA offices in headquarters and regions.
CROMERR Definitions	This document contains a list of special terms used throughout the regulation.

U.S. EPA Disclaimer

Please note that each of the examples listed in this training provides an approach that could help satisfy the associated CROMERR requirement, depending on:

- How it is implemented; and
- How it is combined with approaches to meet the other CROMERR System Checklist items.

Adopting these example approaches does not guarantee that EPA will find the resulting system satisfies the CROMERR requirements.