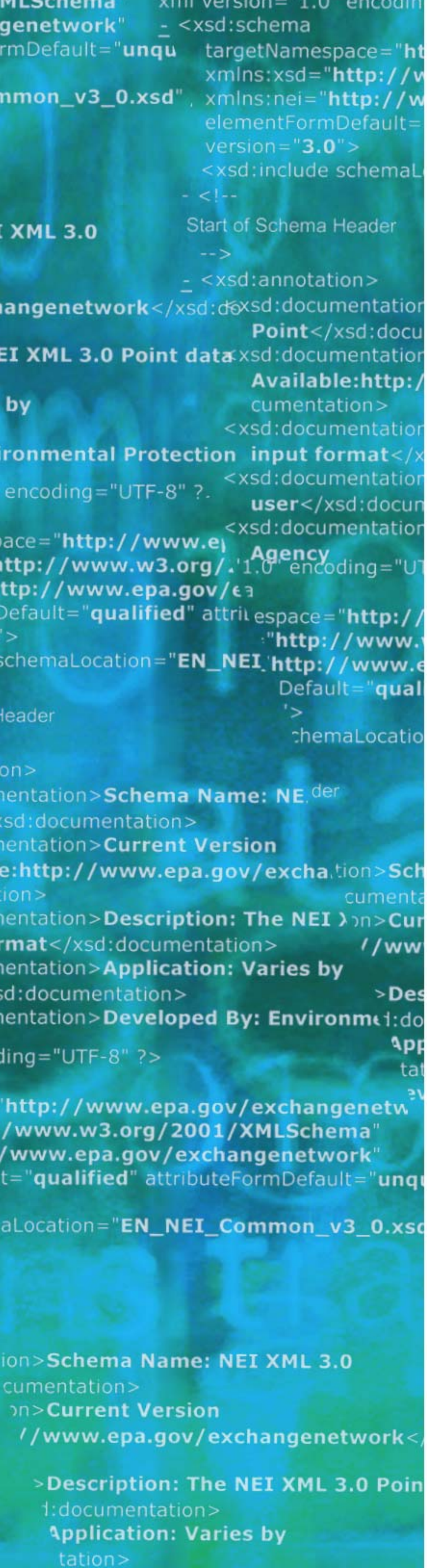

Secure Authentication Key

Version: 1.0

Revision Date: April 6, 2006

Prepared for:
Network Technology Group

Environmental Information
eXchange
Network

Acknowledgements

This document was developed with the input and support of the Network Technology Group

Participant	Affiliation
Connie Dwyer	EPA Office of Environmental Information
Joe Wilson	Office of Water
Chuck Freeman	EPA Office of Environmental Information
Nick Mangus	EPA Office of Air
Chris Clark	EPA Office of Environmental Information
Nathan Wilkes	EPA Office of Environmental Information
Scott Totten	Missouri Department of Natural Resources
Tom Aten	Wisconsin Department of Natural Resources
Dennis Burling	Nebraska Department of Environmental Quality
Dennis Murphy	Delaware Department of Natural Resources and Environmental Control
Randy Moody	North Carolina Department of Environment and Natural Resources
Glen Carr	Oregon Department of Environmental Quality

Additional comments and support were provided by

Yunhao Zhang	
Joe Carioti	CSC
Steve Abercrombie	Ross & Associates Environmental Consulting, Ltd.
Matt Markoff	Ross & Associates Environmental Consulting, Ltd.

Abstract

Secure Authentication keys are a more secure means of authenticating NAAS accounts tied to a specific machine. If widely used, the security of machine-to-machine transactions on the Exchange Network will be enhanced. This document outlines their use, how they should be requested, and explains how they are more secure than current Exchange Network authentication.

1 Secure Authentication Key (SAK)

The common authentication method on the Exchange Network is user name and password. There are a couple of shortcomings with this authentication method:

- Passwords are not strong, and are vulnerable to many attacks.
- When used in machine-to-machine communications, developers tend to embed clear-text passwords in their code. These passwords are easily viewable by others.

The SAK is designed to address common weaknesses in user name/password authentication. A Secure Authentication Key is signed and encrypted using both the user password and NAAS secret key. Similar to a security token, it contains the IP address, but has no timestamp, so it will not expire.

NAAS accepts SAKs as the credential for user and machine authentication on the EN.

An SAK has the following properties:

1. It can be used to replace passwords for all authentications handled by NAAS.
2. It is secured by both the user account password and NAAS secret key. It requires a person to know both secrets in order to create and decrypt an SAK.
3. It is valid only to the machine the SAK is issued to. This prevents it from being stolen and used on another machine.
4. Its integrity is protected. Any changes to an SAK render it invalid.
5. It is spoofing proof. A hacker using IP spoofing would never receive a security token.

SAKs are intended to reduce the exposure of user passwords. Developers don’t have to embed passwords in applications or database tables in order to be authenticated, and passwords do not need to be sent, through public network, to NAAS or network Nodes. In other words, the SAK represents a technique of authentication without presenting secrets.

2 Requesting and Using SAKs on the EN

Secure Authentication Keys can only be created by the Node Help Desk. The Node Help Desk will only issue SAKs to Network Node administrators. SAK requests should only be made for applications or Network Nodes. The following information needs to be provided when a request to create an SAK is made:

- The account information, including account name and credential.

- The IP address of the machine where the key will be used.
- Other additional identity information required by the relying party.

If the request is successfully processed, an SAK key similar to the following will be provided to the requestor:

`key:nCBl8LlDjarXjYJrvGA3A2pPyy8nhmdI5rCsrl96/UY=`

Note that a secure authentication key always has a prefix: ***key***.

2.1 Using Secure Authentication Key

An SAK can be used in both direct authentication (`Authentication` method) or delegated authentication (`CentralAuth`).

The following script shows how to use an SAK in the Node Client object:

```
set myClient = CreateObject ("NodeClient")
'set our SSL certificate file
myClient.SetProperty "Certificate", "NodeClientSSL.pem"
mySecurityToken = myClient.Authenticate ("https://naas.epacdxNode.net/xml/auth.wsdl",
"someone@example.com", "key:nCBl8LlDjarXjYJrvGA3A2pPyy8nhmdI5rCsrl96/UY=", "password",
"")

if mySecurityToken = "" then
        ' we have an error condition
                wscript.echo "Authentication failed - " +
        myClient.GetResponse("Envelope|Body|Fault")
else
        ' The response contains a signature too, show the response with signature.
        wscript.echo mySecurityToken
end if
```

Note that the authentication method is still password in this case.

2.1.1 Use considerations of SAKs

SAKs created by the TEST NAAS cannot be used in the PRODUCTION environment and vice versa. Since SAKs are tied to the IP of a Node server there are several considerations:

- Create separate test and production SAKs
- Don’t use SAKs for client access
- Multiple SAKs may be required if the Node or application that will use the SAK runs on multiple servers

2.2 Secure Authentication Key Construction and Validation

2.2.1 SAK Construction

SAK is a multi-factor authentication mechanism. It requires three pieces of identification information:

- Issuer Identity: an SAK can only be issued by administrators who must present identity information such as `UserId` and `Secret`.
- Subject Identity: The subject the SAK is issued to must have an account with the NAAS, and the account is verified using the account ID and secret.
- Machine Identity: an SAK can only be issued to a specific machine. The machine's IP address is verified.

The server performs the following operations when issuing an SAK:

1. Validate the issuer account and check the secret. Issuing SAK requires administrator's privileges and only authorized administrators can issue an SAK.
2. Construct a string that includes the following information:
   `SubjectId`: The account name to be issued to.
   `MachineIp`: The IP address the token is issued to.
   `IssuerId`: The administrator's account ID.
   `UserData`: Additional user information provided.
3. Attach a digest of the string in the results. This is to protect the integrity of SAK.
4. Encrypt the final string using the server secret and the subject secret using Triple-DES algorithm.

Information in the SAK is very secure. A person would need to know both the user secret password and the server secret in order to decrypt it. Since the combined secret length is much longer than typical passwords, it is extremely hard to crack via brute force, even using the most powerful computer.

As can be seen from the process, an SAK is equivalent to a token signed by three parties: an Issuer, a Subject and the Security Provider. The possibility of obtaining the three credentials at operation time is extremely slim.

The strength of the SAK relies, to a large degree, on the issuer. The ability to create SAKs on the EN will be limited to security administrators.

2.2.2 Validation of SAK

Note that SAK is different from a security token, and it cannot be used as a security token. SAK is a credential issued to a subject; a security token, on the other hand, is an assertion that a subject has been authenticated using a credential. A security token always expires after a predefined period of time, but an SAK doesn't.

When NAAS receives an SAK in an `Authenticate` or `CentralAuth` call, it performs the following operations:

1. Lookup the subject account and verify it is still valid.
2. Pull the secret from the user account and concatenate with the server secret, and then use the combined secret to decrypt the SAK.
3. The SAK is invalid if the decryption failed. Otherwise, the server will verify the digest in the decrypted string to ensure its integrity.
4. Verify that the caller's IP address matches to the IP address in the SAK.
5. Ensure that the SAK has not been revoked.

When *all* the above steps are successful, the server issues a security token signaling the acceptance of the SAK.

2.3 Revocation of a SAK

When a SAK becomes invalid and needs to be removed from the system, an administrator can revoke the SAK by deleting the account associated with the SAK. The SAK will be denied immediately after the account removal.

Partners seeking SAK revocation should contact the Node help desk to request that an SAK be revoked.

3 SAK Security Analysis

SAKs are much stronger than password authentications, and eliminate many of the drawbacks of password-based authentications. In addition, they provide solid defenses to common attacks. One of the key advantages of using SAK is that users do not need to send any passwords cross the network. This dramatically reduces the risk of exposing the secrets.

This section discusses how the SAK technique deals with common attacks.

3.1 IP Spoofing

IP Spoofing is a technical attack in which the sender camouflages a message's origination by using a trusted IP address. IP Spoofing may happen if a hacker steals an SAK and sends it from another machine. SAKs are not vulnerable to IP spoofing attacks.

The security exchange pattern in the Exchange Network is a two-phase protocol. In the first phase, the user is authenticated by sending credentials to the NAAS and is returned a valid a security token. Then, in the second phase, the security token is validated and used for operations. In an IP spoofing scenario, the attacker sends the SAK, but will never receives the security token due to the nature of IP routing (The response will be send to the real IP address, which will fail).

3.2 Replay Attack

Replay Attack is a way of intercepting a message and play it back in its entirely from a different host. Replay attack is the most dangerous attack in password authentication schemes. SAKs can prevent any such replay attacks because it uses both the user and machine identity to authenticate users. Replay attack will fail due to the IP mismatch.

3.3 Dictionary Attack

Dictionary Attack is a way of guessing a subject password by enumerating commonly used passwords.

SAKs are not vulnerable at all from dictionary attacks. Combined server and user secrets are used in encrypting and decrypting SAKs. The secrets are uncommon and would not be easily guessed.

4 Other Security Considerations

- SAKs should only be used for machine-to-machine authentications. It is not recommended for end user authentications.

- When a SAK becomes invalid (due to IP changes on the Server for example), the account associated with the SAK should be removed. This prevents anyone from logging in with the SAK.

5 Conclusions

The Secure Authentication Key technique provides a much better and stronger alternative to traditional password authentications. The deployment of SAK requires minimal changes to the existing network Node authentication systems, but the benefits are significant

The technique is currently being used for connecting backend applications to CDX . It is also being used for supporting single sign-on (SSO) of web applications outside of CDX environments. SAKs can greatly improve the overall security strength in the Exchange Network if the technique is used for all Node-to-Node authentications.