Dynamic Distributed Key Infrastructures (DDKI) and Dynamic Identity Verification and Authentication (DIVA)

André Brisson
Whitenoise Laboratories Canada Inc.
#701 – 1736 W. 10th Ave.
Vancouver, British Columbia Canada V6J 2A6
abrisson@wnlabs.com

***Abstract*—Dynamic Distributed Key Infrastructures (DDKI) is a virtual secure network framework comprised of servers and endpoints that deploy the Dynamic Identity Verification and Authentication (DIVA) protocol.**

Distributed key systems like DDKI are network topologies where network users are pre-authenticated and the keys are pre-distributed to network users. Use of distributed keys eliminates problems associated with key exchange during network sessions.

DIVA is a protocol that is highly distinguishable and novel by enabling

- **One-time-pad encryption**
- **Continuous, dynamic, one-time-pad authentication of an endpoint**
- **Stateful, inherent intrusion detection and automatic revocation without human intervention**
- **Authorization**

DIVA also incorporates two different distributed key handshakes. One hand shake creates a session key for establishing point-to-point communications and enrollment, authentication and activation of new endpoints.

Distributed key systems (like one-time-pad enigma systems) languished because of the requirement for manual distribution of keys. That encumbrance for distributed systems has been overcome. It is now simple, secure and online to create large, distributed authentication and encryption platforms that utilize unique stream ciphers as a distributed private key defined by a unique subkey structure. This enables easy scaling. With this approach there is only partial disclosure of credentials.

DIVA and DDKI can operate as an independent framework but it has been designed to work seamlessly with Public Key Infrastructure (PKI) systems through "The Whitenoise Labs (WNL) Dynamic Identity Verification and Authentication (DIVA) Transport Layer Security (TLS) Extension" [5] or a cipher suite. It hardens PKI and when used with PKI the hybrid systems creates a two channel (asymmetric and symmetric), multi-factor challenge for a hacker where two keys must be broken simultaneously for each and every break attempt. When Whitenoise is used one of the factors a hacker has to break is a dynamic, one-time-pad. "The most striking advantages of Whitenoise are its speed and one-time-pad security that requires the smallest amount of computational effort and resources." [7]

Keywords—DIVA, DDKI, authentication, authorization, encryption, identity, multi-factor

I. INTRODUCTION AND PROCESS

"Technologies now exist to express scalable symmetric key authenticated encryption systems where no single trusted third party knows the final key." [1]

"Robust cryptographic authentication would change the game by employing cryptographic methods which enable secure authentication without transmitting the raw credentials for validation." [1]

Note: For absolute clarity, portions that follow are quoted directly from a patent. These technologies are patented globally i.e. US, China, India, European Union etc.

A distributed key encryption and authentication system and "method is provided in which a key storage server provides a session key to the source and destination computers by encrypting the session key with unique distributed private keys that are associated with the respective source and destination computers by unique private key identifiers. The destination computer then decrypts the encrypted session key using its distributed private key and then decrypts the communication using the decrypted session key." [2]

"Dynamic Identity Verification and Authentication (DIVA) is a robust, real-time identity manager and intrusion detector. Polling sections of highly random but determinist Whitenoise key streams that have never yet been created or transmitted is possible because of offset management. In this context it is not used to encrypt data. It provides one-time-pad authentication. (One-time-pad encryption can be done with the same key.)

"Dynamic Identity Verification and Authentication (DIVA) can be used to provide identity management, continuous dynamic authentication of a user throughout a session (not just at login), immediate hacking detection (the key offsets must remain in sync), and automatic denial of network access (revocation) to hackers and thieves without human intervention.

"These are all significant security capabilities that can be used in new and legacy topologies. The detection capacity is significant. There are no effective, real-time intrusion detection technologies that secure networks throughout entire sessions. It is simple – the offsets between the legitimate user and the server must remain in sync. It inherently detects intrusion or spoofing without human intervention.

"Dynamic Distributed Key architectures as described herein address the aforementioned elements and shortcomings of the PKI system. At the topological level, several network topologies are disclosed that use distributed keys as a random number generator to in turn generate additional distributed keys and securely distribute them to additional devices/persons electronically for easily scalable networks and for scaling secure networks over the Internet. Additionally, these distributed keys can generate session keys for use with any encryption algorithm. Although the preferred embodiment use the keys disclosed in U.S. Pat. No. 7,190,791 (hereinafter "Whitenoise keys" [7]) for additional key generation (and for all security functions including encryption), this may be accomplished with any deterministic random (pseudo random) data source and any encryption algorithms. Adoption of secure network topologies also relies in some contexts on its ability to leverage existing technologies. As such, a hybrid approach is disclosed that uses the Internet's Secure Socket Layer public key technology to add another layer of abstraction to prevent Man-in-the-Middle attacks." [2]

"This invention uses a distributed key, not as a key for a point-to-point link, as would traditionally be done, but instead that key is used to distribute encrypted "session" keys to be used for the original intention of establishing secure links of communication. Distributed keys by their nature, not only allow for the encryption of traffic, but also the authentication of the other party. This is an advantage over the PKI, public key infrastructure, system. [3]

The GateKeeper and Key Vault are application components that are defined in the patents.

"The GateKeeper and the Key Vault work together to create a dynamic distributed key environment for TCP/UDP tunneling. The Gatekeeper creates and encrypts tunnels based on simple standard netfilter rules, while the KeyVault facilitates the retrieval of point-to-point keys as required by GateKeepers as they talk to each other.

"In short, the system currently facilitates near-transparent, dynamic, encrypted point-to-point communication between networks on a network. The KeyVault and GateKeeper systems work together to create a layer on any IP based network, like the Internet, that allows communications to remain secure and confidential." [3]

"The invention provides a dynamic distributed key system. Traditionally distributed key systems require that a key be delivered through courier or in person to each person with whom one wishes to establish a secure link. This invention overcomes this encumbrance. At any time, one can start communicating to someone else that uses the invention without having to wait for a distributed key to be delivered.

"The Authentication Server and Key Vault for the Dynamic Distributed Key Identity Management and data protection system as shown in FIG. 10 have a copy of all physically distributed keys and key pairs for each person/device on the system. The key pairs can be WN-WN, WN-AES, or AES-AES or any other encryption key pairs. The server may have session key generation capacity for creating new key pairs for physical distribution or for encrypted distribution in a dynamic distributed key environment; or, pre-manufactured key pairs can manually be inserted for availability by the authentication and key vault server for additional security and lower processing effort by the server. In a dynamic distributed key environment, new keys are encrypted and delivered to new nodes encrypted in keys that have already been distributed. This eliminates session key distribution using asymmetric handshaking techniques like Diffie-Hellman. Additionally, this model eliminates the need for Trusted Third Parties (outside sources) for the creation and issuance of session keys. Session key generation, when required, is preferably done by the client thereby eliminating this function as a source of increased server overhead. Session key generation may also be done by the server or outside the server by a systems administrator.

"In dynamic distributed key architectures, the server can use its ability to transencrypt the secure traffic through the server from being encrypted in the key of the sender into being encrypted in the key of the receiver. Because of the speed of Whitenoise, it is possible to transcript the entire transmission (file, session keys and vectors) without negative impact on performance. A preferred alternative, to further minimize the computational overhead at the server when using either AES key pairs alone (particularly), or AES-WN key pairs, or WN-WN key pairs, is to simply trans-encrypt the double encrypted session key itself.

A. "Ongoing Identity Authentication Component

"The present system manages the identity of users by 1) initially ensuring that the individual (endpoint) accessing the system is who they say they are, by referencing the last point in the key reached during the last session with the same user (or device). The system stores the point in the Whitenoise stream cypher where the previous session for that user stopped and compares the starting point of the stream cypher at the start of the next session for that user; 2) verifying the user's identity throughout the session; 3) ensuring that a duplicate key is not in existence; and 4) defending the network if an intruder is detected by denying access to both users. The reported loss or theft of a key results in instantaneous denial of access.

"The process provides meaningful and highly differentiated authentication and detection features. The critical insight here is that as content is being consumed, so is the WNkey (Whitenoise key) being consumed. An aspect of the interaction between two end-points is therefore the index into the WNkey. This value is not likely to be known by third parties. Even if the WNkey was stolen, or were the corresponding key structure compromised along with knowledge of the WNL algorithm, ongoing use of the WNkey to gain unauthorized access to protected data would not be possible without the index value corresponding to the authorized history of use between legitimate correspondents. This continuous authentication and detection feature is called Dynamic Identity Verification and Authentication [DIVA].

The DIVA sings only for the correct audience. Not only will illegitimate users of the WNkey be denied, but the legitimate users will immediately and automatically benefit from knowledge of the attack and attempted unauthorized use: the WNkey does not need to be explicitly revoked; it will simply become unusable to its legitimate owner. This can also be accomplished using other non-Whitenoise algorithms that produce long deterministic random (or pseudorandom) data streams or by invoking iterations or serialization of those outputs.

"In the process of ongoing real-time continuous authentication, referred to as Dynamic Identity Verification and Authentication, an unused portion of the key stream is used in a non-cryptographic sense. A chunk of random data from the key (or Random Number Generator) and its offset are periodically sent during the session to the server and compared against the same string generated at the server to make sure they are identical and in sync. This random chunk (unused for encryption) can be held in memory and compared immediately, or written back to media like a USB or a card with write-back capacity for comparison in the future. This segment has never been used and is random so there is no way for a hacker to guess or anticipate this portion of the stream. The unused section of keys stream that is used simply for comparison between server and the client can be contiguous (next section of the key used after encryption), random location jumping forward, or a sample of data drawn according to a function applied to the unused portion of key stream. Whitenoise is deterministic which means that although it is the most random data source identified, two endpoints can regenerate the identical random stream if they have the same key structure and offsets." [2]

DIVA encompasses the following abilities:

"A. Stateful Two-Way and One-Way Authentication

Two-way authentication means that each endpoint can request and send authenticating segments of data or offsets. This means that each endpoint has key generation capability. One-way authentication means that only one endpoint (server/site) has key generation capacity. The server then writes back to the endpoint subsequent segments of key stream data that have not yet been used (and delivers this data chunk securely or otherwise). On the next session, the server/site compares the actual data at the endpoint to the data they can generate using the endpoint's key structure and current offset.

"Currently, authentication of a network user occurs once at login. When an interloper hacks into a "secure" network, the interloper is free to roam around unnoticed. With DIVA, the key stream is polled throughout the session to continually identify and verify that the correct user is on the network. It is possible to incorporate transmission of session keys, use of time stamps etc. to increase the security of initial network access (login) and then DIVA continues to authenticate from there.

"B. Stateful Detection

"The offsets of the key streams must remain in sync between the endpoint and the server. If an interloper manages to steal a key, or gain network access, then the offsets between the server, the legitimate endpoint, and the interloper become out of sync.

There are only two outcomes:

"1) the legitimate owner uses his key/card first and the segment of random key data (or offset) is updated on the legitimate card. The thief then uses the stolen key/card and it won't process because the 1 k data segment (or offset) does not match between the stolen key/credit card and the server. (The hacker has to start over again for next break attempt. No harm has been done.)

"2) The thief uses the stolen key/card first successfully. The next time the card holder uses their card the transaction is refused because the stolen card has been updated with a new offset or segment of data, the offset on the server database has been updated, but not segment of data or offset on the legitimate card. Theft has been identified. The account is immediately disabled. Where the theft occurred is known because of the previous transaction.

"C. Automatic Revocation

"The inherent intrusion detection is simply continuing to monitor that offsets and key segments (tokens) always remain in sync. This is a simple comparison of offset numbers or sections of random data. Without any human intervention, the instant out of sync offsets are detected then the account is frozen and that key is denied network access. It does not require going to outside parties, revocation lists etc. A system administrator can remediate or deal with any situation without worry of continued or ongoing malfeasance

"D. Authorization/DRM

"The assignment and monitoring of permissions and usage rights are accomplished by using different portions of the key stream in the same fashion as authentication." [2]

II. HISTORICAL CONTEXT

The safe exchange of keys is a challenge for electronic and digital communication. Distributed systems stagnated because key management, storage and distribution became onerous. Networks evolved to asymmetric public key systems where session keys were created on the fly. Key storage problems were minimized by having public key databases.

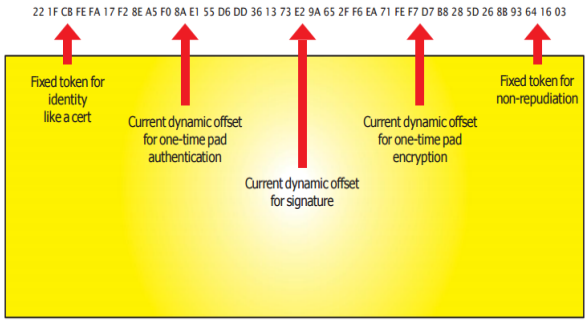
Dynamic distributed key systems are a revolutionary return to the past by using solid, simple, distributed typologies. They are a revolutionary step to the future by offering all key-based security metrics with a single, distributed key when Whitenoise is used.

Historically the number of keys to manage in distributed systems was the square of the number of secure endpoints on a network. Dynamic distributed key frameworks have a one-to-one relationship between the number of keys and endpoints on a secure network.

A single, distributed Whitenoise key creates key streams that are so large that they are very unlikely to ever be exhausted by an endpoint when that key is used as a one-time-pad. Only the internal key structure and the offset (initialization vectors) are required to recreate any key segment. The distributed key structure is represented by a small amount of data describing subkey lengths that make key storage and management easy. For example, "158 bytes of key structure information will generate a random key stream over 100 billion bytes long." [4]

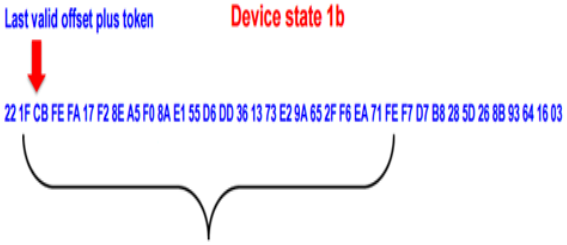
III. DIVA PROCESS

It is helpful to look at a visual representation of this process. The server and the endpoint have a copy of the small key structure that can create very large key streams. When used properly with offsets and initialization vectors, DIVA embeds characteristics of a one-time pad. Static tokens can be used to identify particular services and dynamic identity management tokens for authentication are never used more than once.

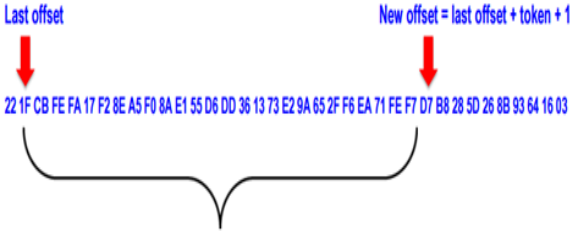


"Both the server and the endpoint have a copy of the account identity management key. The server sends a request to the endpoint device/person for a token of a specific but arbitrary length. Neither an offset nor key is sent with this request." [6]

The endpoint responds by sending a token beginning at the last valid offset of a pre-determined arbitrary length.



The server receives the token from the endpoint and looks up the specific account. It generates a token from its copy of the key of the same length beginning at its last valid offset for that account. It compares the two tokens bit-by-bit. If the tokens are not identical the session is automatically locked. If the tokens are identical the endpoint is authenticated and the session continues.



Continuing the session can be passive since there was no authentication failure. The server can also acknowledge the successful authentication and sends back an authorization to continue. Neither an offset nor key is sent with this authorization.

The endpoint and server update their offsets independently by advancing the offset by the length of the token just generated and compared plus one. The system is synchronized for the next request.

A failed authentication call could only happen if someone was able to copy a key, and use it, prior to a legitimate network access attempt. It is a completely objective process. The keys are either synchronized or not.

There are only two DIVA outcomes. If the legitimate user logs back onto the network first (while a hacker is working away on a break) then the legitimate key and server offset dynamically update and any pirated or spoofed key is no longer synchronized with the server. The pirate would be detected if they make a login attempt. They can't access the network because the pirated key is out of synch. No theft has occurred.

The only other possibility is that a pirate has been successful stealing, copying or breaking a key and logs onto the network first (they have also broken other authentication factors like passwords etc.) In this event, when the legitimate device/user tries to log onto the network subsequently the legitimate key would be out of sync with the server and the account would be locked.

The system administrator then has the needed information for forensics and remediation. The event is contained within

the time frame when the legitimate key was in sync with the server and ended when the account was locked.

The pace of dynamic authentication calls can be arbitrarily set according to security needs and processing contexts. For example, a call can be made with every page change in html or every five seconds if one desired.

The dynamic identity verification protocol can be deployed easily and electronically to any digital device with connectivity, storage and write back capacity. The key structures and initial starting offset is generated by the system. The endpoint requires about 20k of memory/storage. The protocol is started at single-sign-on network access and continues to do dynamic authentication throughout a network session. In many contexts, it can operate without an interface (just inherently) i.e. machine-to-machine communications.

IV. HANDSHAKES

The simplest DDKI handshake involves trans-encryption which converts the data from being encrypted with the key of the sender to being re-encrypted with the key of the receiver in a streaming fashion. This process is conducted at the server which means that the server can potentially capture and compromise communications.

DDKI can also create a handshake where the server (or endpoint) first creates a session key. This is particularly useful in creating point-to-point communications between endpoints. The server will continue to perform dynamic authentication of the endpoints.

Using a session key in this manner can create a secure session with an endpoint that is not on the system yet so that enrollment, authentication and activation of a new endpoint can occur. A one-time distributed private key can be provisioned electronically.

A further configuration is particularly useful with massive networks.

"A preferred alternative, to further minimize the computational overhead at the server when using either AES key pairs alone (particularly), or AES-WN key pairs, or WN-WN key pairs, is to simply trans-encrypt the double encrypted session key itself.

"The trans-encryption process for session keys is as follows. An AES session key is created (preferably at the client). This session key is used to encrypt a file utilizing a standard AES algorithm. This created session key is encrypted with the client's pre-distributed AES private key. This AES encrypted session key is then double encrypted with the pre-distributed AES or WN authentication key (the other key in the distributed key pair) effectively encapsulating and double encrypting the session key and increasing by orders of magnitude the effective security and bit strength of the protection. At the server, the trans-encryption process authenticates the sender by being able to decrypt the authentication layer with a copy of the sender's distributed authentication key, then decrypting the AES session key with a copy of the sender's distributed AES key, then re-encrypting the session key with a copy of the receiver's predistributed AES private key, and finally encrypting all of the above with a copy of the receiver's predistributed authentication key. The double encrypted session key is then embedded in the header of the file and the file is forwarded to the recipient.

"While this is a four-step trans-encryption process, server processing is minimal because only the AES (or WN) session key is trans-encrypted. For example: a 128-bit AES session key is 16 characters or bytes long. The entire trans-encryption process is only manipulating a total of (16 bytes×4 steps) 64 bytes. This is negligible even for strong AES keys. It ensures robust security by strong protection of the session key (never transmitted unencrypted electronically) with minimal server processing." [2]

Interested readers might review a comparison of DDKI handshakes to PKI handshakes which are directly impacted by the overhead related to the manipulation of prime number composites and Dynamic Distributed Key Infrastructure handshakes that are not:

http://www.wnlabs.com/pdf/Comparison_of_handshakes.pdf

V. CONCLUSION

DDKI is a virtual framework that can be used to rapidly harden PKI frameworks. It offers the advantage of relying on a single distributed key that can be used as a one-time-pad. Additionally it offers the advantage of providing identity management, digital provenance with authenticated encryption, dynamic authentication, inherent intrusion detection and automatic revocation of unauthorized network access.

VI. ACKNOWLEDGEMENT

Stephen Lawrence Boren, Vancouver, British Columbia, Canada

VII. REFERENCES

[1] First US National Cyber Leap Year Summit [Online]. Available: http://www.wnlabs.com/papers/National_Leap_Year_Summit_The_Whitenoise_Vision.pdf

[2] A.J. Brisson, "Whitenoise US Patent Number 9166782 granted Dynamic Distributed Key System and Method for Identity Management, Authentication Servers, Data Security, and Preventing Man-in-the-Middle Attacks" www.google.com/patents/US9166782

[3] Albert Meyburgh, "Distributed keys securely sharing session keys" - http://www.wnlabs.com/downloads/Tunnel_Distributed_Keys_distributing_more_keys.pdf

[4] A.J. Brisson,"Cyber Belt presentation for NSA/NIST 2016 – Quantum computing attack resistant security and the cyber belt." http://www.wnlabs.com/pdf/Cyber_Belt_Presentation.pdf

[5] Laurie Perrin and A.J. Brisson, "TLS Whitenoise DIVA extension", http://www.wnlabs.com/technology/WNL_TLS_extension.php

[6] A.J. Brisson, "Visual Look at the Whitenoise Algorithm" [Online] http://www.wnlabs.com/pdf/WhitenoiseAlgorithmVisualLook.pdf

[7] A.J. Brisson, "Deterministic One-Time-Pad Generation – Making a Whitenoise Key" page 1 [Online] http://www.wnlabs.com/pdf/ATC_IEEE_TrustedComputing_creating_a_WN_key_2017.pdf