Whitenoise Security as a Service

Whitenoise Security as Service uses distributed keys to provide secure network access with dynamic, continuous, one-time-pad (password) authentication throughout a network session. With perfect identity and unique authenticated encryption provenance is attributed to persons, endpoints and data.

Dynamic Distributed Key Infrastructures and Dynamic Identity Verification and Authentication also establish point-to-point encrypted tunnels and can create and distribute session keys for dynamic, secure point-to-point connections for other network areas and uses such as key distribution.

Server Features

- Store, create, distribute and manage Whitenoise keys
- Create dynamic session keys and securely distribute unique private and link keys once for real-time enrollment of new persons, devices and networks
- Perform continuous dynamic one-time-pad authentication of all endpoints

One distributed key

- Maintains continuous identity management and provenance
- Performs all network security controls
- Is unbreakable

Benefits

- One distributed key performs all network security controls i.e. secure network access, continuous-dynamic-one-time-pad authentication through a session, authorization, signature, intrusion detection and automatic revocation.
- Prevent all known/anticipated cyber attack classes like man-in-the-middle and quantum.
- It is easy to implement and understand, fast, and requires no training.
- It is interoperable and scalable for real-time enrollment of new endpoints and networks.
- Secure identities can be restored and refreshed for victims i.e. Target, Home Depot etc.

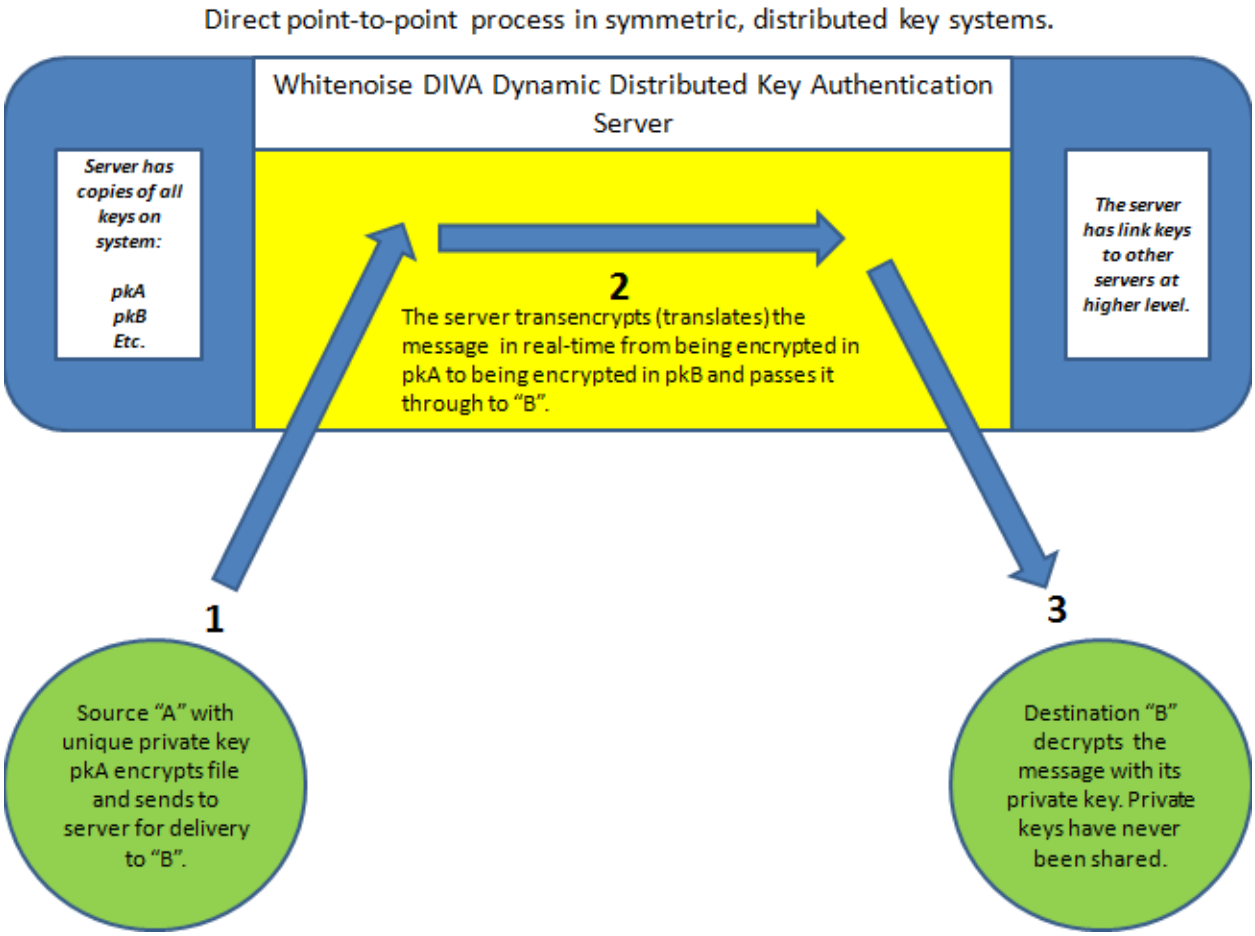
You can choose to eliminate user names and passwords altogether.

Comparison of PKI and DDKI Handshakes

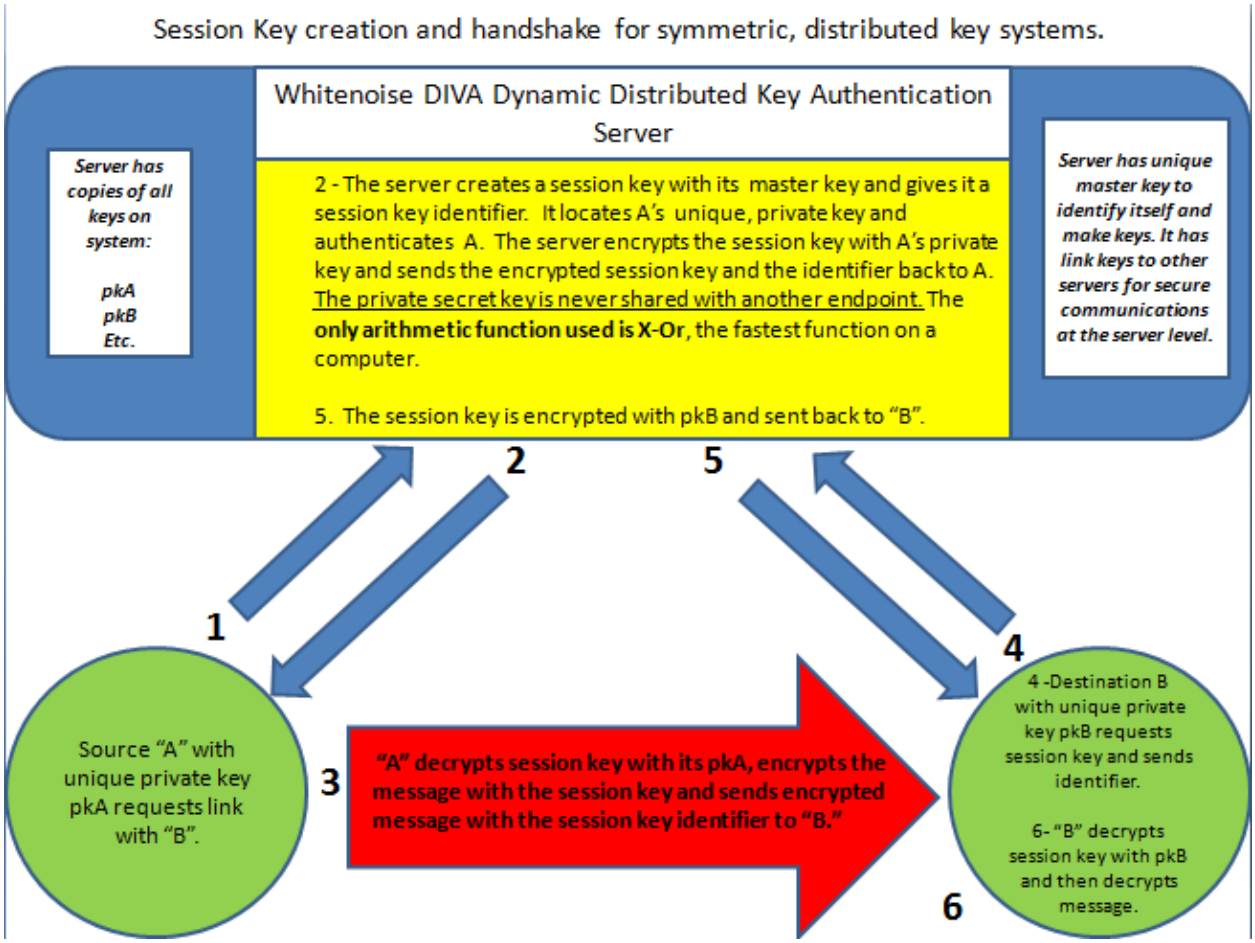
Rules:

In symmetric, dynamic, distributed key systems the server has copies of all the keys on a system. The keys are stored in an encrypted state. The keys are always kept separate from the last current dynamic offsets.

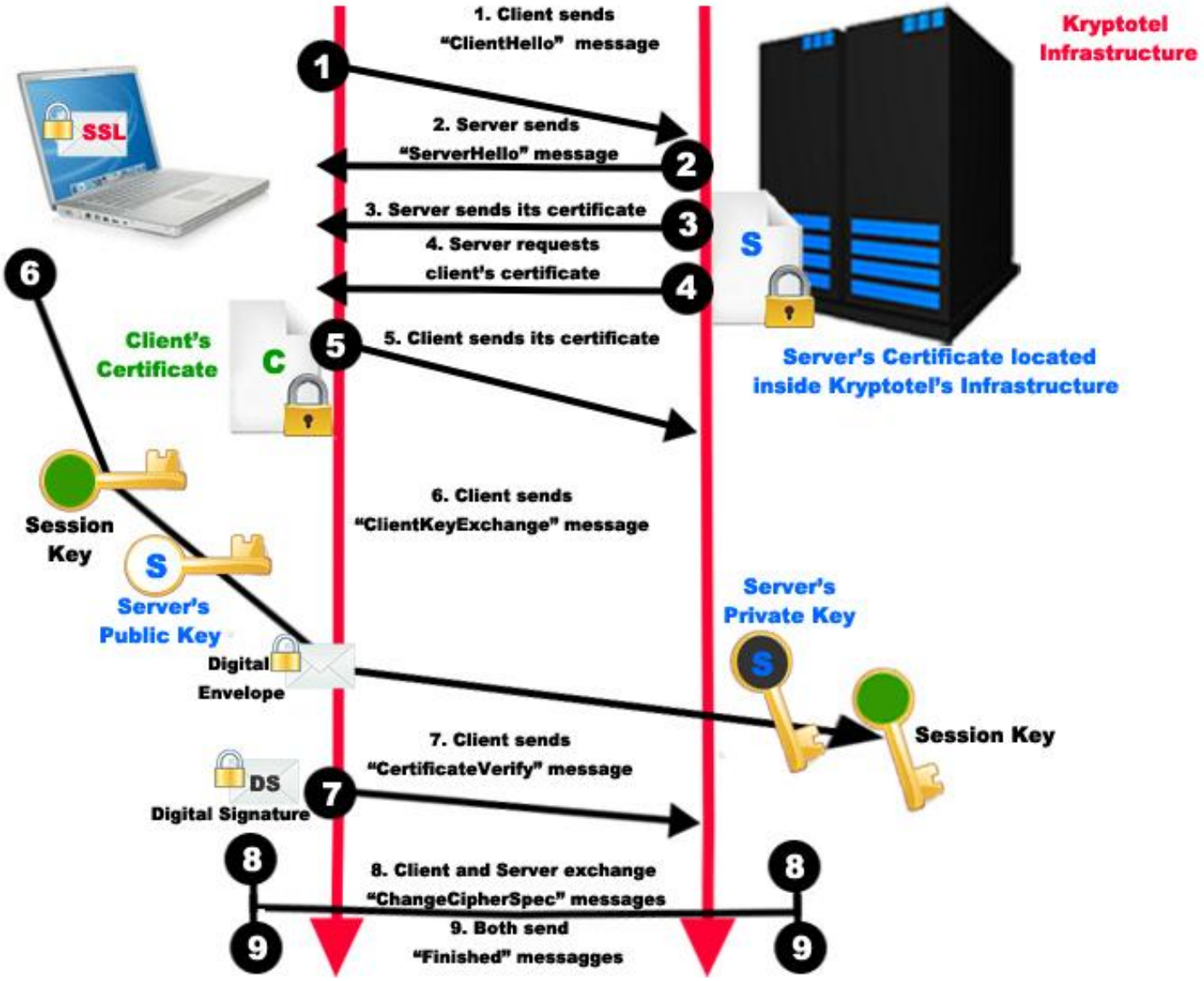
Each endpoint has only its unique, distributed, private/secret key. Secret keys are NEVER shared between endpoints. There is never key or offset exchange after setup. The following illustration shows a system in its simplest configuration.



It is desirable to be able to generate session keys in order to communicate with endpoints that do not yet have their own private/secret key. The purpose is to be able to establish secure communications with a new endpoint without first having to copy a key physically to that endpoint as has been traditional in distributed key systems. This allows simple scaling of the network. It facilitates authentication and secure, one-time key distribution. It facilitates the establishment of a secure point-to-point connection between endpoints without intercession (trans-encryption) by the server. The server continues to dynamically and continually authenticate the endpoints but no encrypted traffic is passing through it for potential capture.



For comparison, let's look at the complexity of a typical asymmetric, public key process and key exchange handshake.



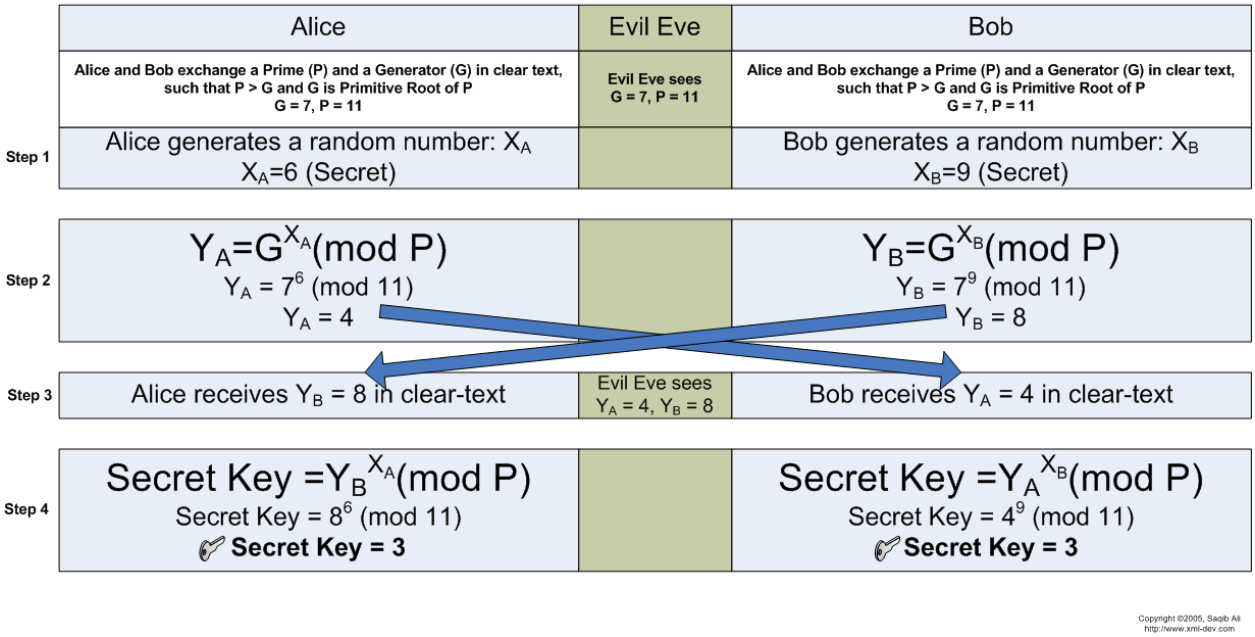
Its complexity is further aggravated by the very intensive mathematics required by PKI. It requires so much overhead in power, space and computational resources that it is literally unusable in many environments including the Internet of Everything where the majority of networked components operate under restrictive environments.

This is one mathematical technique for creating an asymmetric, public key session key. By comparison, after key load, the only operation used by DIVA and DDKI is X-Or. The either-or function is the fastest function available on a computing device.

whitenoise technologies

TELECOM COUNCIL 2015 SPIFFY AWARD NOMINEE

Diffie Hellman Key Exchange



The security of public key systems has always been fatally flawed. They can never prevent Man-in-the-Middle attacks and a host of other attack classes. In the past, computers were so slow that none of the attacks were considered feasible. Now computers are so fast it is simple to ravage our networks because of lack of sustainable identity and data provenance and easily compromised security processes.

Dynamic Distributed Key Infrastructures (DDKI) and Dynamic Identity Verification and Authentication (DIVA) prevent all known and anticipated cyber attack classes:

- **Man-in-the-Middle attacks** are prevented because there is no key exchange.
- **Side Channel attacks** are prevented because all operations are order 1 after key load and because there is no access to the key.
- **Mathematical and factoring attacks** are prevented because keys are created by a binary mechanical process as opposed to arithmetic ones requiring multiplication and mods.
- **Botnet attacks are prevented** by configuration with server so the botnet never has access to all the key material to authenticate data being sent OUT of a network or computer.
- **Brute force attacks** are not feasible with the continually changing dynamic offsets.
- **Denial of service attacks** can be prevented by exploiting unbreakable identity and a proxy for secure network access so that hackers could never get on a network.
- **Quantum computing attacks** are prevented because every variable is variable.