Tactical Crypto Assertion Warning

Tactical Asymmetric / Symmetric Cryptography.

If you've been referred here, please take a moment to review this.

These tools are provided as a way to create compatibility to some customer defined cryptography use cases. These tools are low level cryptography primitives and **shouldn't** be used without clear understanding of the risks associated.

For many use cases, there is far better tooling already in the product that enables customers to set up strong security, without building it "by hand" with these policy elements.

For instance, the most common use cases that we've heard about are better served by existing well supported features:

- To secure parameters in an API call, SSL is far easier to implement than these tools, and is actually more secure.
- To provide encrypted credentials, JWT, JWE, JWS are good choices.
- For encrypted message bodies, JWE comes to mind as well.
- Older standards like SOAP, WSS, SAML do a lot of the same things as JWT, JWS, JWE.

Instead of implementing these assertions, please consult with Tactical and senior presales staff on how to get the feature the customer needs.

The entire deployment: source, destination, configuration and the tooling used to implement security is sometimes referred to as a cryptosystem.

In nearly every customer use we know of for these assertions, the cryptosystem **as deployed** lacks current state of the art protections against cryptographic attacks.

Specifically, well-scrutinized systems like SSL and WS-Security BSP have mechanisms for establishing a session key used for the symmetric cryptography, normally via the Diffie-Hellman Key Exchange: https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

Without a key derivation/symmetric key refresh, both Asymmetric and Symmetric cryptography is often vulnerable to cryptographic attacks. Some symmetric ciphers are subject to https://en.wikipedia.org/wiki/Known-plaintext_attack , and all symmetric crypto is weak to password derived keys – there's too little entropy, making it vulnerable to a brute force https://en.wikipedia.org/wiki/Ciphertext-only_attack

Asymmetric crypto is too expensive for bulk encryption, in terms of CPU, so that's usually not a good choice for performance reasons.

Another risk is especially present is if the tools are used without user authentication/authorization. With cypher block chaining algorithms, there's a particularly large possibility of creating a key oracle via the padding oracle attack. See: http://robertheaton.com/2013/07/29/padding-oracle-attack/

For field deployments, we're very reluctant to release this, because of the huge numbers of possible ways to get it wrong.