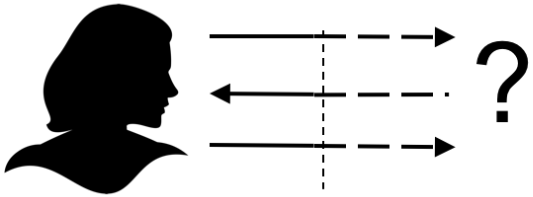
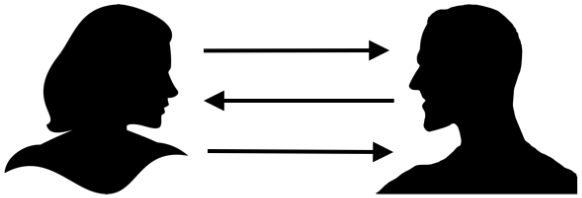
CPSA

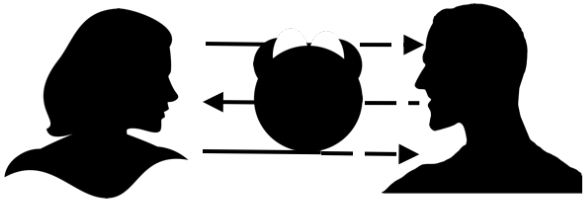
The Cryptographic Protocol Shapes Analyzer is a tool developed by the MITRE Corporation to assist protocol designers to analyze cryptographic protocols to find critical defects that can enable attacks such as man-in-the-middle attacks.



Alice, a protocol participant, can only observe the messages she has sent and received so far



The simplest explanation may be for Bob, her intended communication party, to have a *matching* experience



CPSA can automatically identify how an adversary may manipulate the situation so that Bob has a *non-matching* experience

Cryptographic Protocol Shapes Analyzer

Protocol designers need to adapt or design cryptographic protocols to support a wide variety of security goals for the Internet of Things (IoT), cloud-based architectures, vehicle-to-vehicle (V2V) communication, etc. The Cryptographic Protocol Shapes Analyzer (CPSA) is a MITRE-developed software tool that assists in the design and analysis of such cryptographic protocols.

The design or modification of such cryptographic protocols is notoriously difficult. Even highly-scrutinized, standard protocols, or minor adaptations, have been shown to be flawed in subtle ways. Therefore, CPSA is extremely valuable in helping to eliminate an entire class of attacks against the protocols that secure our networks and device interactions.

It can automatically discover a wide class of network-based attacks, and it can prove the security of protocols against such attacks. By visualizing the variety of protocol executions in the presence of an adversary, CPSA helps boost intuition about why a protocol succeeds or fails to achieve its security goals.

Key features

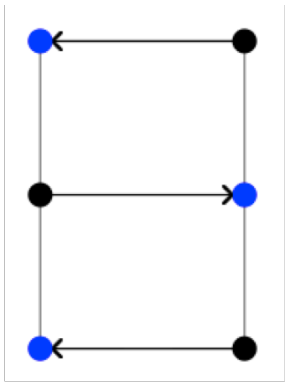
- simple and intuitive
- strong formal foundations
- mature and supported
- independent of protocol implementation
- open source

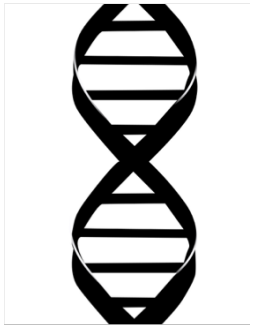
Simple and intuitive

Designers describe protocols from the point of view of each protocol participant. CPSA works by enumerating a set of executions (shapes) of a protocol. These are all the essentially different explanations—accounting for a network adversary—for some observed behavior. Although there is no limit to the number of executions consistent with the given behavior, there are frequently very few shapes, often just one or two, even when the protocol is flawed.

If there is an attack against the protocol, it is represented by one of the shapes. It is easy to "read off" from the shapes the authentication and secrecy properties the protocol satisfies. An important advantage of computing the shapes is the ability to explore the protocol design space in the absence of rigorous security goals.

CPSA is unique in supporting a quick and visual comparison of alternative design choices. Our team has successfully trained numerous new users with little or no knowledge of cryptography. Self-guided training materials are also included in the distribution.





Based on the formal theory of *strand spaces,* named after the resemblance of protocol diagrams to DNA strands

Formal foundations

CPSA is the result of years of sponsor-funded MITRE research. It is based on the formal theory of strand spaces—so named due to the visual resemblance between message diagrams and strands of DNA. The theory behind CPSA's algorithm has been subjected to intense scrutiny through many peer-reviewed publications. This provides protocol designers with very high assurance of the correctness of CPSA's conclusions.

Mature and supported

MITRE has used CPSA to evaluate and improve the design of group keying protocols proposed for small unmanned aviation systems (SUAS). CPSA also helped MITRE design a remote attestation protocol (called CAVES) for securely reporting a target machine's state without revealing its persistent identity. MITRE has engaged with standards bodies to provide expert commentary, improving the quality of proposed protocol standards.

CPSA continues to be at the heart of MITRE research into protocol security. CPSA has several advanced features that reflect an evolving understanding of how cryptographic protocols interact with the systems around them. This includes support for protocols involving a stateful hardware security device such as a Trusted Platform Module (TPM), and methods for measuring protocol strength using security goals. The user interface is also adapting to the evolving needs of the growing community of CPSA users.

Implementation independent

Since CPSA aims to find design flaws that do not exploit weaknesses in the underlying cryptographic primitives, it uses a so-called Dolev-Yao, or symbolic, model of a network adversary. Intuitively, this model captures the types of attacks that could be performed by a compromised router in the path between protocol participants. The adversary sees every message sent on the network, and can choose to redirect, reflect, replay, copy, drop, or alter messages in transit. It can also perform cryptographic tasks, such as encryption and decryption, assuming it has access to the relevant secret keys.

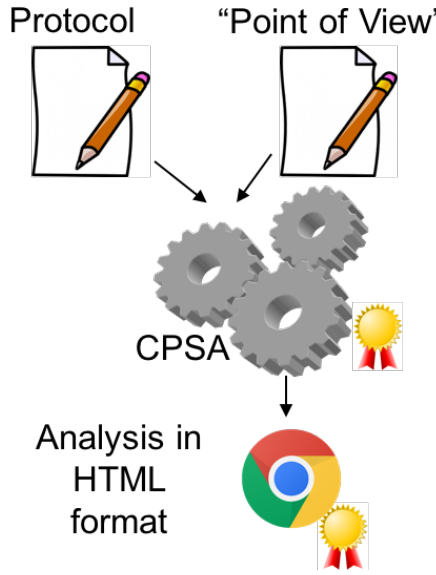
CPSA supports abstractions of symmetric and asymmetric encryption, digital signatures, and cryptographic hashes. It has provisional support for protocols based on a simple Diffie-Hellman key exchange. CPSA cannot discover cryptanalytic attacks or vulnerabilities in protocol implementations. Since adversary knowledge of secrets is all-or-nothing, it does not capture password guessing attacks.

CPSA is not tied to any layer of the OSI model. Since it is based only on the underlying logic of the cryptographic primitives, CPSA can analyze cryptographic protocols at any layer.

Open source

CPSA for Linux, MacOS, and Windows, is open source and freely available at: https://github.com/mitre/cpsa/.

CPSA requires Haskell Platform which can be downloaded for free from: https://www.haskell.org/platform.



Minimal requirements

Designers only need a text editor, a web browser and the Haskell Platform. Designers specify the sequence of messages each protocol participant sends and receives. This defines the formal model CPSA uses when searching for shapes. The designer then specifies a "point of view," which is typically a description of a single local session, that is, the experience of one of the protocol participants. The point of view contains important assumptions, such as which long-term keys are not initially known to the adversary and which values are randomly chosen.

CPSA processes the inputs and outputs its results in HTML format. The designer then inspects the output, looking for any anomalous executions.

MITRE