Belief Logic is a process by which we can analyze protocols in a logical manner. We are not so much looking to prove these protocols secure; instead we wish to show that our authentication goals have been achieved. The symbols and constructs that we will use are listed below.

The symbols $A$, $B$ and $S$ denote specific principals. Principals can be people, computers or services. $K_{AB}$, $K_{AS}$, $K_{BS}$ denote symmetric secret keys shared between $A$ and $B$, $A$ and $S$, and $B$ and $S$, respectively. $K_A$, $K_B$, $K_S$ denote the public keys of $A$, $B$, and $S$, while the inverses of these keys (e.g. $K_{A^{-1}}$) represents each principal's private key. The symbols $N_A$, $N_B$ and $N_S$ identify nonces as well as their creator. $X$ and $Y$ are statements or messages. The following constructs are used to show the usage and relationship for all principals, keys and statements.

$P \models X$: $P$ believes that $X$ is true.

$P \mathrel{\triangleleft} X$: At some point in time (past or present) $P$ received some message $X$

$P \mathrel{\triangleright} X$: At some point in time $P$ sent $X$. Also, at the time of sending, $P$ believed $X$.

$P \mathrel{\Rightarrow} X$: $P$ has jurisdiction over $X$, meaning other principals believe $X$ if they believe $P$ believes $X$.

$\sharp(X)$: $X$ is fresh. $X$ has not been sent at any time before the current run of the protocol. Nonces are expressions generated to prove freshness, and often include a timestamp. Without nonces, it is possible to get that not so fresh message feeling.

$P \mathrel{\leftrightarrow} Q$: $P$ and $Q$ share the key $K$ and may use it to communicate. Furthermore, $K$ will never be discovered by any principal except $P$, $Q$ or a principal trusted by $P$ or $Q$.

$P \mathrel{\rightarrow} K$: $P$ has public key $K$. The private key $K^{-1}$ will only ever been known to $P$ or principals trusted by $P$.

${X}_K$: Represents $X$ encrypted under key $K$. When $K$ is a private key (e.g. $K_{A^{-1}}$) this represents a signature of $X$.

1. Message meaning rules concern the interpretation of messages. Rather than using the new symbols, we will write the English equivalents.

When using shared keys,

\[
\frac{P \text{ believes } Q \mathrel{\leftrightarrow} P \quad P \text{ has seen } {X}_K}{P \text{ believes } Q \text{ once said } X}
\]

When public keys are used,

\[
\frac{P \text{ believes } \mathrel{\rightarrow} Q \quad P \text{ has seen } {X}_{K^{-1}}}{P \text{ believes } Q \text{ once said } X}
\]

2. *Nonce-verification* rules show how to check that a message is fresh, and that the senders believes so as well:

\[
P \text{ believes } X \text{ is fresh, } P \text{ believes } Q \text{ once said } X \quad \quad \quad \quad \quad \quad \quad P \text{ believes } Q \text{ believes } X
\]

3. The *Jurisdiction* rule states that a principal *P* will trust the beliefs that *Q* has jurisdiction over.

\[
P \text{ believes } Q \text{ controls } X, \quad P \text{ believes } Q \text{ believes } X
\]

\[
P \text{ believes } X
\]

4. A principal that sees a formula in plaintext, also sees its components:

\[
P \text{ sees } (X, Y) , \quad P \text{ believes } Q^K \rightarrow P, P \text{ sees } \{X\}_K
\]

\[
P \text{ sees } X, \quad P \text{ believes } K \rightarrow Q, P \text{ sees } \{X\}_{K^{-1}}
\]

\[
P \text{ sees } X.
\]

Note that even if *P* sees *X* and *P* sees *Y*, then *P* does not necessarily see *(X, Y)*.

5. If any given part of a formula is fresh (and the formula cannot be altered), the entire formula must be fresh:

\[
P \text{ believes fresh}(X)
\]

\[
P \text{ believes fresh}(X, Y).
\]