

Silo Watching

by
David L. Chaum

Consider the problem faced by a nation that wishes to monitor sites in a host nation. The monitoring nation wants a judge (e.g. the U.N.) to be able to determine the authenticity of reports made by sensors at each of the monitored sites. But the host will co-operate only if it can be sure that the monitoring nation can not determine which sensor makes which report.

One way to cast the problem is:

- (1) The monitor, host, and judge can authenticate and read reports created by each of n sensors.
- (2) The monitor, host, and judge can determine how many sensors they are receiving reports from.
- (3) The host can keep the correspondence between the reports and the sites from which they are issued secret—even if the cryptosystem is subverted.
- (4) A captured sensor whose information content is compromised by the monitor will not reveal the correspondence between the reports issued by other sensors and the sites from which they are issued.
- (5) The monitor can keep the host from forging reports made by sensors that have not been compromised by the host.
- (6) The host can keep the monitor from forging reports made by sensors that have not been compromised by the monitor.

A possible solution, that is based on public key cryptography, involves the following steps:

- (1) The host forms a list of public keys H_1, H_2, \dots, H_n ; the monitor forms a list of public keys M_1, M_2, \dots, M_n . Both lists are supplied to the judge, and to the other nation.
- (2) The monitor presents its corresponding list of private keys to each sensor i , $1 \leq i \leq n$. These lists are protected in transit to the sensor by being encrypted with the sensor's public key S_i . That is, the lists received by the sensor are of the form $S_i(M_1^{-1}, M_2^{-1}, \dots, M_n^{-1})$.
- (3) The monitor may now destroy its own copy of the list of private keys, so that the list does not fall into the wrong hands.
- (4) Each sensor i issues a copy of the pre-arranged constant C that has been signed by the keys in the list received in step (2) and its own private key: $S_i^{-1}(M_1^{-1}(\dots M_n^{-1}(C)\dots))$. After checking each of these, the host provides them to the monitor and judge. This ensures that each sensor received all the private keys, $M_1^{-1} \dots M_n^{-1}$.
- (5) The host supplies each sensor i with a different secret integer σ_i , $1 \leq \sigma_i \leq n$, and the private key $H_{\sigma_i}^{-1}$, corresponding to the public key H_{σ_i} of step (1).

- (6) Each sensor i destroys all keys except $H_{\sigma_i}^{-1}$ and $M_{\sigma_i}^{-1}$, so that if it is compromised later by the host, it will not contain keys sufficient to allow reports from other sensors to be forged.
- (7) Sensors will issue reports periodically, say once a day, based on the time supplied by an internal clock. Sensor i signs the report R_d for day d with the key supplied by the host and then with the key from the list supplied by the monitor: $M_{\sigma_i}^{-1}(H_{\sigma_i}^{-1}(R_d, d, C))$. After checking each day's reports, the host forwards them to the monitor and judge as a single batch ordered on σ_i .
- (8) Upon receiving a day's batch, the host and judge use the public keys M_i and H_i to authenticate the signature of and read the i th report.

The question of who trusts what about the sensors themselves can be divided into the following concerns:

- (1) The monitor and judge want to be assured that the host can not compromise an active sensor to obtain $M_{\sigma_i}^{-1}$, or influence the construction of sensors to cause them to make false favorable reports.
- (2) The host and judge want to be assured that the monitor can not compromise an active sensor and thereby obtain σ_i or $H_{\sigma_i}^{-1}$.
- (3) The host and judge are concerned that the monitor may build into the sensors a capacity to fail or provide unfavorable reports at some point during the sensor's operation so that the host may be discredited and/or a small amount of information may be leaked by the timing of the degeneration.
- (4) The monitor and judge are concerned that the host may falsely claim that the monitor has carried out some threat described in (3) above.

Concerns (1) and (2) above are easily dealt with. The monitor builds the sensors and equips them with a capability to destroy their own information content if they are tampered with. The host secures the site of each sensor so that an attempt to compromise the sensor by the monitor will cause the sensor to be destroyed.

A solution to threats (3) and (4) requires that the host and monitor (and perhaps the judge) agree not only that the sensors will operate properly, but also that they are not likely to fail.