Payoff

Fixed passwords are no longer appropriate for controlling computer access. Effective access control calls for the use of dynamic passwords, which are generated by tokens, a calculator-type device. Many such devices have now been introduced into the marketplace, but no one is necessarily appropriate for all situations. This article discusses the use of dynamic passwords and describes the characteristics of currently available password generators and their advantages and disadvantages in particular situations. A chart comparing the features of a selected group of tokens is included.

Problems Addressed

A popular method for gaining computer access is the fixed password, which is a secret set of characters that are assigned to the user. Increasingly sophisticated computer abuse, however, has made it clear that this form of personal authentication is not secure. Some of the problems of fixed passwords are that they are:

- Able to be monitored, trapped, copied, and replayed over communications lines and on networks.
- Too often copied or written down.
- Often too simple and easily guessed.
- Frequently inadvertently disclosed when initially assigned or handed out.
- Too often shared.
- Able to be learned by observing the user keystrokes.
- Often subject to trading by hackers.
- Too easily obtainable by decoy or deception.
- Changed too infrequently (and too frequently changed back to the original).

As a result, dynamic passwords—generated once by a special calculator—are becoming a preferred method for access control. These password generators, or tokens, pose new configuration and administration challenges, however, which are discussed in this article.

Dynamic Passwords

The dynamic, or one-time, password is becoming a popular alternative to the fixed password. The basic concept of dynamic passwords is to prove a person's identity by testing to ensure that that person possesses a unique key or password generator. The user is provided with a special-purpose calculator that generates an unpredictable number. This number is then used as a one-time password to enter into the computer. In some cases, the number is produced unilaterally by the token; in others, it is calculated in response to a challenge from the computer. In all cases, for each requested entry, the security software or hardware installed at the access control point calculates an expected response to the one-
time password calculated by the token. If the two numbers match, the security system grants computer access to the individual carrying the token.

Because a new password is produced or calculated each time access is requested, no password need ever be written down or memorized. Even if a password is copied, it is useless because a new password is generated each time. In some systems, reuse of the same password by the same user is extremely unlikely but statistically possible. Other systems offer protection against all reuse. In newer product offerings, the token may also be used as an employee ID card, a physical access control device, a calculator, or a credit card.

There are two essential parts to token-based access control—the unique tokens issued to the authorized users, and the access control security system (software or hardware) installed at the access control point. (See the following section for a further discussion of authentication architectures for use at access control points.)

A user gains computer access by entering a unique user ID into the access control system through the terminal or workstation. The access control system evaluates the user ID and determines whether it is authorized and if so, how user authentication should occur—through a fixed password, a dynamic password, or in some cases, a biometric device.

For dynamic-password authentication, the access control system database contains the type of token and the unique seed, or cryptographic key, stored in the token for each user ID. Other information about that user is also stored in the access control system, including authority group, location, fixed passwords, and authorization controls. Most access control systems have addressed the problem of lost tokens or unauthorized use of legitimate tokens. If an authorized user's token is lost, the user cannot access the system and would therefore report the token as missing. The computer security administrator simply deletes the information on the prior token and replaces it with new data on the replacement token. To prevent unauthorized use, most systems use a PIN to activate tokens. Without the proper PIN, the token still provides a password, but an incorrect one. Some tokens also provide duress codes, so that the security software can recognize when users are being forced to use the token and issue appropriate warnings.

### Authentication Architectures

Five security architectures are currently available for access control and user authentication.

#### Workstation Authentication

This approach, sometimes called the Peripheral Defense, places the authentication and access control system in the workstation. Normally boot protection is also provided. Essentially, a user cannot gain access to the workstation, nor to its ability to gain access to other resources, without first proving that the specific requesting user is entitled to have such access. Generally, all workstations that have the capability to access the protected target resources must have the authentication capability.

#### Dedicated Authentication Systems

These are generally freestanding hardware devices, installed in front of the computer resources to be protected. They are designed to protect access to the protected resource and also generally offer such nonsecurity capabilities as menuing and session logging.

#### Access Server Authentication

These systems are general-purpose communication devices, with various control, user menuing, and routing/switching features, to which security and authentication functions have been added.
Host-Based Authentication
These software systems are designed to be installed on the protected resource itself to control access at the first entry port level or host communications point-of-entry. On large mainframes the access control and authentication functions are usually coupled to the functions of a resource control program (e.g., Resource Access Control Facility, ACF2).

Authentication Nodes
This type of system offers an authentication server for the entire network. Either operating under Kerberos or a single sign-on approach, the user authenticates only once and is either given a ticket allowing access to other network resources, or is granted access to a set of macro auto-logon script files which can then be used to obtain access to other resources on the network.

Modes of Token Operation
The two most common modes of operation are asynchronous and synchronous. In the Asynchronous mode, the access control software issues a cleartext challenge to the user that is displayed on the terminal or workstation screen. The user turns on the password generator, enters a PIN, enters the cleartext challenge into the token, and presses a key to cause the token to calculate the response. The response is then displayed on the token and the user keys that value into the terminal or workstation. Because the access control software in the protected computer knows the unique seed (i.e., encryption algorithm) assigned to that user's token, it can calculate the expected response. If the two responses match, the user is granted access.

In the synchronous mode, the access control software requests the password without calculating and presenting a challenge to the user. The user turns on the password generator, enters a PIN, reads the response from the display, and keys that value into the keyboard of the terminal or workstation. The computer knows the expected response through a combination of three factors: it knows the algorithm the token uses to calculate the response; it knows the unique key assigned to that token that will be used in calculating the response; and it knows the method used by the token to maintain dynamic password synchronization with the access control system. Maintaining password synchronization is a key factor in synchronous tokens. Asynchronous tokens essentially are resynchronized each time they are used, since the access control system issues a new challenge on each use. Synchronous tokens essentially issue their own challenge and the access control system must be able to determine what that challenge is. There are three common methods to do this: time-synchronous, involving use of time and other factors (using the clocks in the token and in the access control system, and allowing for clock drift); event-synchronous, involving use of a value developed from one-time modification of the last entry; and algorithmic-synchronous, involving reverse engineering of the response to determine if the specific token could have generated that response. As in the Asynchronous mode, if the two responses match, the user is granted access. As in the Asynchronous mode, if the two responses match, the user is granted access.

Passing Authentication Between Computers
In addition to the conventional use of tokens, there are five variations in using authentication which are important to consider. These are:

- Workstation-to-host authentication.
- Workstation single sign-on.
· Network authentication nodes.

· Host-to-host authentication.

· Host-to-user authentication.

In certain applications it may be desirable to authenticate the workstation, rather than the individual user. This is generally the case where the workstation is in a secured area and may be used by multiple people. Sometimes the use of tokens is not acceptable or cost-justified. In these cases a noncopyable software token may be installed in the workstation. (This approach obviously will not work with dumb terminals.) The user or system administrator will be required to authenticate at boot-up, generally with a fixed password. Subsequently, any access request that is challenged will be answered automatically by the software token, transparently to the user. In cases where dynamic password security is used, no password aging is required. Otherwise, the user or the software token must be able to respond to requests for password aging from the host system.

An important variation of the software token is the use of a single sign-on software module in the workstation. For the user who has need to access multiple resources that require authentication (even if only ID and fixed password), single sign-on should be considered. This module works exactly like the software token, but has the capability to store multiple software tokens and/or logon macro script files. As with the software token, the module is noncopyable and requires workstation authentication to be activated. (Token authentication at the workstation level is highly recommended.) Once activated, the module will automatically respond to authentication requests from any protected resource for which the module has an entry.

An important development is the evolution of the network authentication node of which there are two types:

· **Kerberos Authentication Nodes.** This type of node is being actively developed by a number of companies working to support this public domain software, and by numerous user organizations. In this approach, the user logs into the Kerberos node (with either a fixed or dynamic password) and after authentication is given an encrypted, time-stamped “ticket” that he can then take to any resource controlled by a Kerberos access module, present the ticket, and if the ticket is correct, gain access. There is only one database, no synchronization is required, and access is available from any workstation. However, there is no session control and no logging of complete sessions.

· **Session Control Nodes.** With this type of node, the user logs into the authentication node and after authentication is given a menu that contains that specific user’s choices for system or resource access. When he makes his selection, the authentication node automatically logs the user into the requested resource and remains present during the entire session. This approach allows for the authentication node to provide the communication pathway to each resource and stay with the user during the entire session, providing complete session control and logging. When the user completes that session or is logged out of that system, he is once again presented with his menu by the authentication node. There may be only one database, or multiple databases. It is therefore possible to have several authentication nodes to balance the communication load. The functioning of the authentication node is an integral part of the regular Network Operating System and protocols, therefore access to the authentication node is available from any workstation.

To date, a limited amount of work has been done with host-to-host(also called peer-to-peer) authentication (except in the area of Electronic Data Interchange). However, there is
rapidly growing interest in this capability, and it is not difficult to implement. The access control system can be installed as a gateway access system, a system utility in the host (generally as part of the normal logon procedure), or can be software imbedded in an application program that is used in the peer-to-peer process. The responding system (essentially a software token or a secure autolog script file) can be installed as part of the telecommunications access software, or can be imbedded in the application program that requests the peer-to-peer process. Note that it is not the user that is being authenticated here, but rather the host or host application. It is probably wise to have the user who initiates the process authenticate himself to the system or application to enable use of the peer-to-peer authentication process.

Host-to-user authentication has a limited purpose—to assure the user that the correct host has been accessed. This prevents simulating the intended host and trapping the user access to obtain IDs and passwords.

Types and Characteristics of Tokens

A wide range of token devices are on the market. Most are asynchronous, using full challenge and response. All have some form of encryption, ranging from the full Data Encryption Standard (DES) to a variety of proprietary algorithms. Some tokens are calculators, most are not; some have replaceable batteries, some are disposable after the batteries wear down (usually within three to five years). Smart cards are now being developed for use as tokens with both hard-wired and portable readers. Some smart cards and tokens can store multiple seeds and synchronization information that enable the user to access more than one computer without having to enter a long, random challenge. Some have the ability to operate with multiple encryption algorithm in multiple modes. All are easy to use and carry. The following sections describes some of these characteristics and their advantages and disadvantages.

Initialization

Some tokens are initially programmed at the factory, with the unique key inserted or developed before shipment. Many tokens, however, are shipped blank, and the data security administrator must do the programming. (Generally, factory-programmed tokens can be ordered at an extra charge.) Although blank tokens may be more work for the data security administrator, they are often considered more secure than preinitialized tokens, which could be compromised between shipment from the factory and receipt. On the other hand, if the keys are developed under factory control, the data security administrator cannot compromise the tokens.

To eliminate both these concerns, some tokens are designed to be field initialized by the end user. This type of token can be securely initialized even if the initialization is carried out across an unsecured network. Such cards were originally designed for online information services providers to be sent out through the mail to remote users, who would then log onto the system and initialize their card by themselves. Only after this secure initialization process is completed can the privileged security supervisor gain access through the security software to the unique key. The user may reprogram the card at any time. This type of token was designed to provide direct accountability for system use. When users log onto the online system, they must prove their identity to gain access and, unless a token is reported lost or stolen, they are then held accountable for the resulting bill for services.

When tokens are not initialized at the factory, the method of programming the tokens must be decided. Manually programming a few tokens is fine and may be necessary for some remote sites. Programming hundreds of tokens, however, is tedious and time-consuming. An automatic programming device is recommended when tokens are not programmed at the factory.
Physical Characteristics
Tokens are available in five basic physical types:

- A hand-held calculator type with replaceable batteries.
- A flat calculator-type card without replaceable batteries (sometimes referred to as a supersmart card).
- The conventional smart card with a chip embedded in the card, usually accompanied by a hand-held device into which the card is slipped to provide the keyboard and display.
- The software token described earlier.
- A hardware device without a keyboard or display, manually installed by the user on a dial-up line, programmed to automatically respond to access control security system challenges.

Two main issues related to the physical characteristics of tokens are user friendliness and alternative applications of the token. User friendliness is of particular concern to organizations issuing tokens for the first time, especially to outside customers or to senior managers. They want to have a token that is unobtrusive and very easy to carry and use. Some of the newer tokens can be used as an employee ID card, a physical access control device, a calculator, or a credit card. Opinions differ on whether tokens should be single-use devices (emphasizing the importance of security) or multiple-use devices (increasing user friendliness).

Keyboard and Display
All of the devices come with a form of liquid crystal display, and most have a numeric keyboard. Some have keys for clearing the display and backspacing, making it easier for the user to correct mistakes in entering the challenge or in programming the card.

In both Europe and the United States, the introduction of the credit-card type smart card has brought about the need for a hand-held device into which the card can be inserted to provide for standard token operation. (Normal use of these type of tokens is with a cable-connected card reader.) These hand-held devices have battery power, a keyboard, and a display, but rely on the smart card itself for memory and processor capability.

Three modes of display are commonly offered in the most popular tokens—straight decimal, hexadecimal, and a modified, nonambiguous Hexadecimal. Some of the characters used in Hexadecimal display have the potential of being confusing (e.g., the number 6 and the lower-case letter b). Users who have problems with this display mode should be given tokens that use a straight decimal or nonambiguous Hexadecimal mode, which substitutes ambiguous characters with nonconfusing characters. Hexadecimal displays provide greater security, however, because of the greater number of combinations that can be represented.

A final point about the display regards automatic shutoff, which is offered with most cards. This feature conserves battery power and reduces the exposure of information on the card display.

Maximum Password Length
The longer the response to a challenge, the greater the security. This is simply a function of the complexity and time required to crack an encrypted response. At some point, however, additional security is infeasible or uneconomical in light of the marginal gain that it provides. The maximum password length for two of the cards in Exhibit 1 is 16 digits. (It could have been higher in those tokens but was limited to 16.) In the other tokens, the limit is seven or eight. These limits are built into the tokens themselves, rather than in the
supporting software. The chances of guessing a dynamic eight-digit password are one in 108, a large enough number to discourage most intruders.

**Token Comparison Chart**

**Minimum Challenge Length**
The challenge is used only in Asynchronous tokens. The supporting software controls the challenge length. Many security supervisors reduce the size of the challenge to improve ease of use. In some of the tokens a soft PIN (discussed in a later section) is used, which can also be used to reduce the number of characters in the challenge or eliminate it.

**Synchronous Host Support**
If a user is working on more than one computer, secure access can be ensured by:

- Using multiple tokens, one for each resource.
- Placing the same unique key in the data base of each of the supporting software systems. This solution, however, could compromise the secrecy of the key because it is the same on each machine; therefore, the security of each system depends on all of the others.
- Using a different PIN or password for each machine where the PIN or password is combined with the one-time response.
- Using a token that has the ability to support multiple keys.
- Using a software token or single sign-on module that employs Asynchronous token technology (full challenge-response) that is transparent to the user in use.

If, however, a synchronous mode of operation is used and each computer has a different synchronization factor, the token must have multiple synchronous host support; that is, the token must be able to keep track of the synchronization factor for each machine. This is relatively easy for time-dependent tokens because of the clock in each machine and in the token control synchronization. The software must allow for clock drift between the two clocks to be in synchronization (current systems do so). The primary risk in the drift allowance is exposure of the password; during the time in which the password's validity is being confirmed it must be protected so that it cannot be used on other resources under the same software system. With event-synchronous tokens, on the other hand, the token must be able to keep track individually of the last event for each computer used. Without that capability, accessing a different computer causes the synchronization to be changed, destroying the synchronization for the previous computer and necessitating that a full challenge and response sequence be performed to reestablish synchronization. Algorithmic-synchronous tokens have neither of these problems.

**Hard Versus Soft PINs**
Two types of PINs are used in tokens—hard PINs and soft PINs. A hard PIN is entered into the token by the user and is evaluated in the hardware of the token logic; it is not known or evaluated by the software in the host computer. Therefore, the hard PIN need never traverse a network nor be entered into the host computer software. A hard PIN can
be changed in the token without coordinating that change in the host computer. Data security administrators have minimal control over hard PINs.

A soft PIN is entered into the token by the user and directly influences the way in which the dynamic password is calculated. Unlike conventional fixed passwords, the soft PIN never traverses the network and is never directly entered into the host system by the user. The host computer software evaluates the dynamic password to determine whether the user entered the correct soft PIN. Therefore, a change in the soft PIN in the token must be coordinated with the host computer software, usually by the data security administrator.

The use of either type of PIN is highly recommended by the token vendors because unauthorized users cannot use a token to gain access without knowing the PIN. Hard PINs are usually programmed into the token at the factory; some can be changed in the field. Soft PINs are generally set up by the factory or the data security administrator but are then changed at once by the user with a utility that interacts with the user and the host software. The utility software reverse engineers the soft PIN to determine the new PIN using constants known to both the token and the utility software.

There are differences of opinion regarding which type of PIN is best. Hard PINs are much simpler to administer. Soft PINs are much more flexible and can provide an additional level of security. Some tokens support both hard and soft PINs.

When deciding whether to use a hard PIN, the data security administrator should consider the following factors:

- Whether a token accepts a hard PIN or whether a hard PIN is optional.
- The PIN size. A larger PIN is harder to break, but a four-digit PIN is considered standard and in most cases offers adequate security.
- Whether the hard PIN can be changed in the field.
- Whether the token has an attack deactivation. This last feature disables the card after a certain number of wrong entries. It can be a desirable feature for foiling unauthorized users.

The key factors in evaluating soft PINs primarily deal with whether support exists in the host security software and the size of the PIN supported. It is assumed that soft PINs are always user changeable.

Encryption Algorithms

Three types of encryption are used in tokens to calculate unique dynamic passwords:

- The DES (DES). The application of DES to tokens does not carry the strict export restrictions imposed by the US government, because DES is used here to encrypt only the passwords, not user data.
- ANSI X9.9. A one-way encryption variant of DES that is primarily used in message authentication.
- Proprietary algorithms.

A discussion of the advantages and disadvantages of various algorithms is beyond the scope of this article; company policy often dictates which algorithms may be used and therefore which tokens may be selected.

It should be pointed out that encryption used in token authentication is not subject to export controls as are encryption systems for use in encoding user data. Since the only thing that is being encrypted and decrypted is the one-time password, the US Government
does not restrict export of token technology. Smart cards that have encryption algorithms and cipher storage capability are subject to export controls.

**Operation Mode**
As discussed previously, there are two main modes of token operation—asynchronous and synchronous. The Asynchronous mode always uses a challenge and reponse, and the synchronous mode does not use the challenge. Some tokens offer both modes, some only one. The buyer must carefully consider the environment that is to be secured and the characteristics of the user community before choosing an operation mode.

The following six factors may influence token selection:

- Asynchronous tokens require more keystrokes than do synchronous tokens and are therefore considered less user friendly.

- There are no synchronous tokens with replaceable batteries. (Some buyers prefer not to use throwaways).

- If only software tokens are used, there is no advantage to using synchronous tokens.

- Synchronous tokens may require additional administration by the security administrator for token/system synchronization.

- Users may already have tokens from another environment or application that can be used in the new environment.

- In multiple host environments there may be administrative or security issues that can be avoided with asynchronous tokens.

**Battery**
All hand-held tokens run on batteries. Batteries are evaluated according to their lifetime, whether they are replaceable, and whether the token must be reprogrammed when the battery is replaced. Batteries that are not replaceable should be guaranteed for long life. If the batteries are replaceable, it is preferable that the token need not be reprogrammed when the batteries are replaced. The major disadvantage of replaceable batteries is that access into the token case must be provided; because of this need to provide access as well as the battery's bulk, the cases must be larger than they are for tokens that have nonreplaceable batteries. Many users prefer smaller cases. Size preferences must be weighed against the cost of replacing the entire token when the battery dies.

**Warranty**
The standard warranty for the tokens is now generally one year, and the tokens have proved to be quite reliable.

**Keystroke Security Ratio**
The keystroke security ratio is the number of keystrokes required to generate a password with a token, using a four-digit PIN, that reduces the possibility of guessing the correct password to less than one in one million. Exhibit 1 includes the keystroke security ratio for various tokens. Tokens that operate in the synchronous mode have an advantage in that there are no keystrokes required to enter the challenge. Token keyboard controls also play a role in that on buttons and enter keys can add to keystrokes. The point of applying this ratio is to gain the best balance between user friendliness and adequate security.
Product Offerings
There have been several implementations of token technology using the smart card format. AT offers a smart card system with both a portable reader and an adjunct reader coupled to the user workstation. The portable reader is equipped with a full keyboard. When the user inserts the smart card into the reader and turns it on, the unit functions just like a conventional challenge-response token. With the adjunct reader the user inserts the smart card, performs the initial host login, and the challenge-response is done automatically by the unit, transparently to the user, thereby eliminating the keystrokes. The AT smart card uses Data Encryption Standard and has its own storage and directories, PIN security, and message authentication capability. Up to four secret keys may be programmed for each of eight different host systems. A similar system is offered by ThumScan.

A portable, pocket-sized device for remote authentication is offered by LeeMah DataCom Security Corporation. The InfoKey works in conjunction with the LeeMah TraqNet security system. It is about the size of a cigarette package and is easily jack-plugged into the line by the user, who then uses his or her workstation or laptop to log into the assigned TraqNet system. When TraqNet has verified the selected host, it issues a challenge to the user that will be automatically answered by the InfoKey for the user. The user does not have to do any key entry of either a challenge or a response.

Several vendors are now offering software tokens, including Digital Pathways, LeeMah, and Enigma Logic. These tokens are software modules installed on the user workstation, rather than hand-held hardware devices carried by the user. They function exactly as does a challenge-response hardware token, but eliminate the need for the user to carry a token or to key in challenge or response data. Because the software token is a valuable item, it must be properly secured to prevent people other than the authorized user from copying or removing the module. Also, since it must be installed on the workstation being used by the user to access the secured resource, it is normally installed only on that one workstation and is not moved from one workstation to another.

Recommended Course of Action
With the increasing use of networks and of outside access to computer resources, the need for security has never been greater. Authentication is the keystone in a sound security program. Based on knowledge of who the user is (with a high degree of certainty), we can control access, authorize the user privileges to perform functions and manipulate data, allow the use of encryption/decryption engines, and log and effectively hold the user accountable for his or her actions. Without effective authentication, these functions cannot be performed with certainty. Dynamic password technology, whether implemented in hardware tokens or in software is a sound, secure, and reliable way to obtain effective authentication.

Security administrators responsible for selecting tokens should evaluate vendor offerings on the basis of cost, ease of use, level of security, and industry or corporate standards, with each factor weighted according to its importance to the organization. The host system security software should also be selected with as much care as the token to achieve optimal security.

Author Biographies
Joseph T. Hootman
Joseph T. Hootman is president of Computer Security Systems, Inc., a computer and information security consulting and product sales firm based in Northern California.
<table>
<thead>
<tr>
<th>COMPARISON CRITERIA</th>
<th>VENDOR</th>
<th>Safeworld MultiSync</th>
<th>Safeworld AccessCard</th>
<th>Racial WatchWord</th>
<th>Secure Net-Key</th>
<th>Safeworld DES Gold</th>
<th>Safeworld DES Silver</th>
<th>Sec Dynamics SecurID Card</th>
<th>Sec Dynamics SecurID Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td></td>
<td>–</td>
<td>–</td>
<td>RG500</td>
<td>SNK004</td>
<td>–</td>
<td>–</td>
<td>SDY520</td>
<td>SDY200</td>
</tr>
<tr>
<td>Hard PIN Support</td>
<td></td>
<td>Optional 0.2-8</td>
<td>No</td>
<td>Required 4-8</td>
<td>Required 4-16</td>
<td>Optional 0.2-8</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PIN size</td>
<td></td>
<td>Yes</td>
<td>n/a</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>User Changeable</td>
<td></td>
<td>Optional</td>
<td>Optional</td>
<td>No</td>
<td>No</td>
<td>Optional</td>
<td>No</td>
<td># Option</td>
<td>No</td>
</tr>
<tr>
<td>Token Attack Deactivation</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Optional</td>
<td>No</td>
<td># Option</td>
<td>No</td>
</tr>
<tr>
<td>Soft PIN Support Available</td>
<td></td>
<td>Optional</td>
<td>Optional</td>
<td>No</td>
<td>No</td>
<td>Optional</td>
<td>No</td>
<td># Option</td>
<td>No</td>
</tr>
<tr>
<td>Encryption Algorithm</td>
<td></td>
<td>DES/ ANSI X9.9</td>
<td>Public Key/ Proprietary</td>
<td>DES/ Proprietary</td>
<td>ANSI X9.9</td>
<td>DES/ ANSI X9.9</td>
<td>DES</td>
<td>Proprietary</td>
<td>Proprietary</td>
</tr>
<tr>
<td>Operation Modes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronous</td>
<td></td>
<td>Event</td>
<td>No</td>
<td>No</td>
<td>Event</td>
<td>Algorithmic</td>
<td>No</td>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>Asynchronous</td>
<td></td>
<td>Optional</td>
<td>Yes</td>
<td>Yes</td>
<td>Optional</td>
<td>Algorithmic</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Battery Replaceable</td>
<td></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No up to 4 years</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Battery life</td>
<td></td>
<td>3 years</td>
<td>3 years</td>
<td>2 years</td>
<td>3 years</td>
<td>3 years</td>
<td>3 years</td>
<td>Life, # option</td>
<td>Life, # option</td>
</tr>
<tr>
<td>Warranty</td>
<td></td>
<td>1 year</td>
<td>1 year</td>
<td>90 days</td>
<td>1 year</td>
<td>1 year</td>
<td>1 year</td>
<td>Card life</td>
<td>Card life</td>
</tr>
<tr>
<td>Price for single unit</td>
<td></td>
<td>$30-40</td>
<td>$20-30</td>
<td>$57-85</td>
<td>$50</td>
<td>$40-50</td>
<td>$39-48</td>
<td>$42 and up</td>
<td>$34-70</td>
</tr>
<tr>
<td>Initialization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factory</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Security Supervisor</td>
<td></td>
<td>Yes</td>
<td>Yes (TP)</td>
<td>Yes</td>
<td>Yes (TP)</td>
<td>Yes (TP)</td>
<td>Yes</td>
<td>Yes (TP)</td>
<td>Yes (TP)</td>
</tr>
<tr>
<td>User (TP=Trusted Person)</td>
<td></td>
<td>Yes</td>
<td>Yes (TP)</td>
<td>Yes</td>
<td>Yes (TP)</td>
<td>Yes (TP)</td>
<td>Yes</td>
<td>Yes (TP)</td>
<td>Yes (TP)</td>
</tr>
<tr>
<td>Automated Programming</td>
<td></td>
<td># Option</td>
<td>No</td>
<td># Option</td>
<td># Option</td>
<td># Option</td>
<td># Option</td>
<td># Option</td>
<td># Option</td>
</tr>
</tbody>
</table>