ARX (Algorithmic Research) PrivateServer Hardware version 4.7 Firmware version 4.8.1

FIPS 140-2 Non-Proprietary Security Policy

Level 3 Validation

April 2012

Table of Contents

1 INTRODUCTION

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Algorithmic Research PrivateServer. This security policy describes how the PrivateServer meets the security requirements of FIPS 140-2, and how to operate the PrivateServer in a secure FIPS 140-2 mode. This policy was prepared as part of the level 3 FIPS 140-2 validation of the PrivateServer.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 -- *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the NIST web site at http://csrc.nist.gov/cryptval/.

1.2 References

This document deals only with operations and capabilities of the PrivateServer in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the PrivateServer and other Algorithmic Research products from the following sources:

- Algorithmic Research web site contains information on the full line of security products at www.arx.com.
- For answers to technical or sales related questions please refer to the contacts listed on Algorithmic Research site at www.arx.com.

1.3 Terminology

In this document the Algorithmic Research PrivateServer is referred to as the module or the PrivateServer.

1.4 Document Organization

The Security Policy document is part of the FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Machine
- Module Firmware Listing
- Other supporting documentation as additional references

This document provides an overview of the PrivateServer and explains the secure configuration and operation of the module. This introduction section is followed by Section 2, which details the general features and functionality of the PrivateServer. Section 3 specifically addresses the required configuration for the FIPS 140-2-mode of operation.

With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Submission Documentation is Algorithmic Research-proprietary and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Algorithmic Research.

2 FIPS 140-2 security level

PrivateServer is validated to meet the FIPS 140-2 security requirements for the levels shown below. The overall module is validated to FIPS 140-2 security level 3.

FIPS 140-2 Security Requirements Section	Level
Cryptographic Module Specification	3
Cryptographic Module Port and Interfaces	3
Role, Services and Authentication	3
Finite State Model	3
Physical Security (Multi-Chip Standalone)	3
Cryptographic Key Management	3
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	N/A
Operational Environment	N/A

Table 1 – FIPS 140-2 Security Requirements Level

3 The PrivateServer

The Algorithmic Research PrivateServer is a high-performance cryptographic service provider. Contained within a secure, tamper-responsive steel case, the PrivateServer performs high-speed cryptographic

The Algorithmic Research PrivateServer is a high-performance cryptographic service provider. Contained within a secure, tamper-responsive steel case, the PrivateServer performs high-speed cryptographic operations while protecting sensitive data. All keys and critical security parameters are protected within the cryptographic boundary by the physical security mechanisms of the module.

The PrivateServer supports various cryptographic algorithms including AES for encryption and SHA-256 for hashing. It can be used to securely store secret/private keys and has the ability to maintain an internal public key database. The PrivateServer performs all cryptographic operations internally, and through self-tests it ensures that these operations are functioning correctly. There is no room for error when protecting mission critical data.

Whether performing the backend cryptography for a high-volume e-Commerce site or just providing authentication services for a small company, the PrivateServer satisfies the need with its wide-range of cryptographic functionality. It includes the following features:

- Cryptography using Triple-DES, AES, Triple-DES-MAC, HMAC, Triple-DES-CMAC, AES-CMAC, AES-CCM, RSA, ECDSA, SHA-1, SHA-256, SHA-384 and SHA-512.
- Public key database and certificate support
- Authenticated and encrypted communication with the module
- Secure storage of secret/private keys
- Software key medium and smartcard support
- Tamper-responsive enclosure
- High level API requiring no cryptographic expertise
- In-depth logging, auditing and secure auditing
- Secure backup capabilities
- Code Mailing capabilities

3.1 Secure by Design

The PrivateServer is a multi-chip standalone module. PrivateServer hardware version 4.7 with firmware version 4.8.1 has been designed to meet all of the Level 3 FIPS 140-2 requirements. This means that the module provides strong security both inside and out.

Encased within a tamper-responsive and tamper-evident steel box, the module both protects against and reacts to attacks.

All vents on the module are baffled meet the FIPS 140-2 opacity requirements for physical security.

Access to the module is only permitted through specific, well-defined interfaces detailed in the following section (3.2).

The security features of the module ensure that access to sensitive information is granted only to an authorized operator. Tamper Evident cans provide evidence of any attempt to tamper with module cover. The Tamper Evident cans are placed over a screw that joins the top cover and bottom enclosure.
The Tamper Evident cans are applied at manufacturing stage.
The Tamper Evident cans are shown in Figure 1.

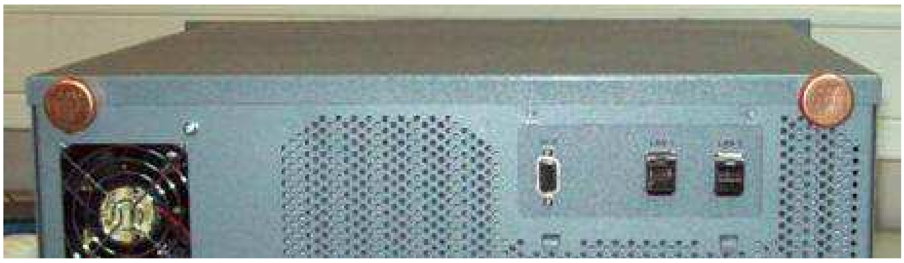


Figure 1 – Tamper Evident cans

The units are encased in a solid metal case rigged with micro-switches and only the specified physical interfaces permit access to the module. Intrusion attempts cause power to be instantly cut off, preventing access to any useful information by zeroizing all plaintext critical security parameters including the PrivateServer Critical keys.

Two smartcards (Init Smartcard and Startup Smartcard) are used for the purpose of initializing the module. The initialization must be done in a secure environment.
The above PrivateServer Critical Keys are split between these two smartcards. Thus it is mandatory to insert both smartcards for a successful initialization of the module.
During initialization of the module, a part of the PrivateServer Critical Keys is kept inside the internal Tamper Device. Therefore, for a normal startup of the module, it is only required to insert the startup smartcard.
After a detected tamper, the PrivateServer must be re-initialized with both Init and Startup smart cards.

Remark: It is possible to configure PrivateServer such that there is no need to enter the startup smartcard as part of starting the module. In this configuration all PrivateServer Critical keys content is kept inside the internal tamper device and erased upon a tamper event.
This can be done using the PrivateServer console.
Use the *Unattended Mode* option in the PrivateServer's console to configure this option. You will need to insert the startup smartcard as well as enter the startup smartcard password. This will enable the unattended startup configuration.
The same console option can be used to use back the attended startup configuration.

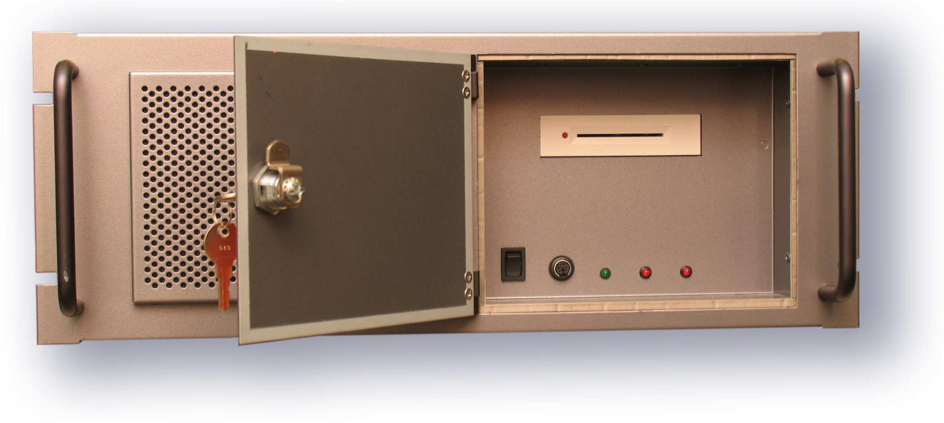
All services of the module must be carried out through a secure session.

The module meets FCC requirements in 47 CFR Part 15 for personal computers and peripherals designated for home use (Class B). It is labeled in accordance with FCC requirements.

3.2 Well-Defined Ports

The module is a hard, rack mountable box. The physical ports include the power connector, network connections (Ethernet Interfaces using TCP/IP and UDP/IP), power switches, indicators, a monitor port, a keyboard port, and one smart card reader. The module is encased in a steel cover, with only the specified ports providing access to the module. All ports use standard PC pin outs.

The ports are shown in Figure 2. On the front of the module behind the access door you have a smart card reader in the top middle. Below that, from left to right, you have an on/off button, keyboard port, and three indicator lights. On the back of the module, left to right, you have the power connector and power switch below the fan and the monitor port and two network connections on the top right. These ports are all listed in table 2.



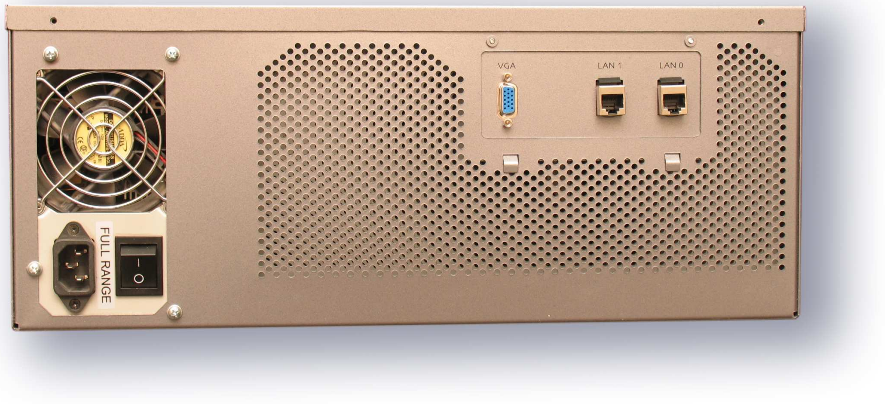


Figure 2 – Front and Rear Interfaces

For FIPS 140-2 purposes, both network ports are treated the same. Through the network ports either an encrypted and RSA based authenticated sessions are permitted or a user ID/password authenticated sessions are permitted over either ports when operating in a FIPS 140-2 compliant manner. In a non-FIPS 140-2 compliant manner, the module could be configured so that traffic over the secure Ethernet port was plaintext while traffic over the unsecure network was encrypted and authenticated or user-ID/password authenticated.

Table 2 shows the mapping of the FIPS 140-2 logical interfaces to the module's physical interfaces.

FIPS 140-2 Logical Interfaces	**Adapter physical interfaces**
Data Input Interface	Network ports, keyboard port smartcard reader
Data Output Interface	Network ports
Control Input Interface	Network ports, keyboard port, buttons
Status Output Interface	Network ports, indicators, monitor port
Power Interface	AC power connector

Table 2 – Interfaces

All requests for cryptographic services are done through the PrivateServer API. This API, written primarily in C and based on RPC (Remote Procedure Calls), provides a high-level interface to the cryptographic services provided by the module, thus masking many of the complexities of cryptography from the developer. Figure 3 depicts this API model.

Status information can also be sent via syslog protocol to a syslog server or via SNMP traps to an SNMP server. This status information is sent using the network ports of the module.

A special license can enable PrivateServer printing codes to a network printer. A client application will interface PrivateServer through the normal data input interface. The application will either send the code encrypted or direct the PrivateServer service to calculate a new code for the given user. The code will be sent non-encrypted to the network printer that will use a special paper for concealing the code.

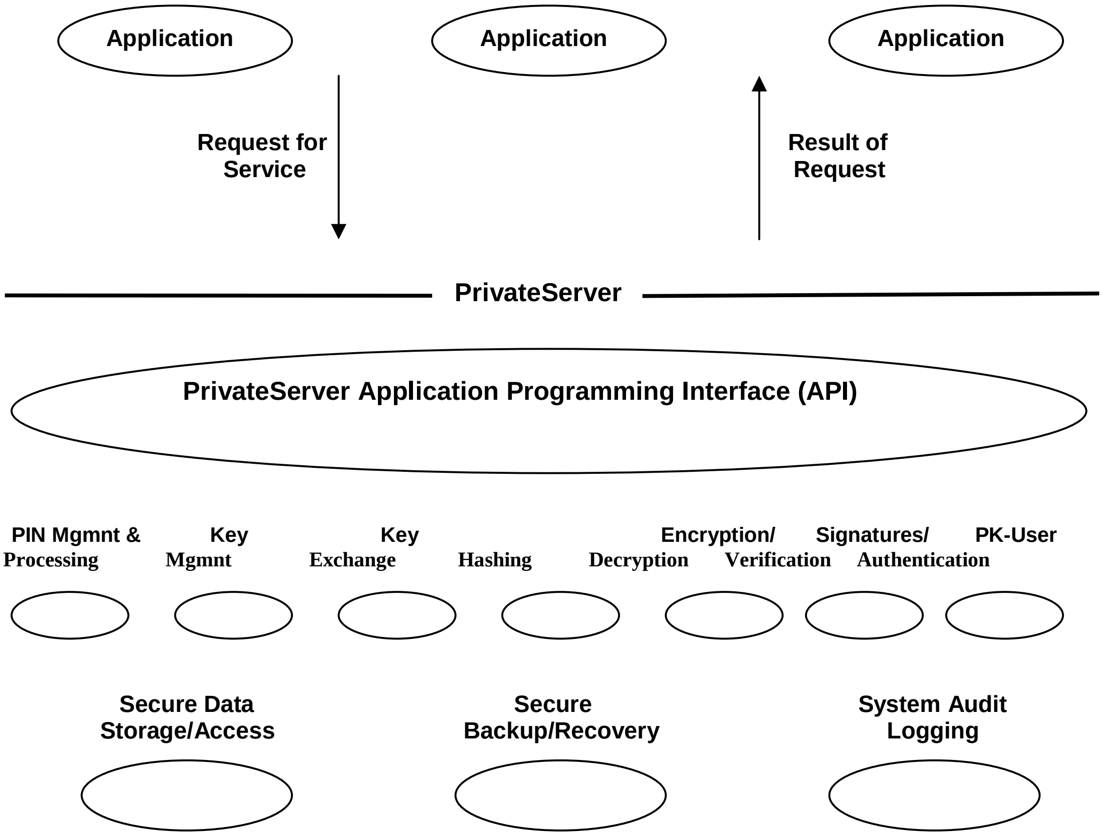


Figure 3 – PrivateServer API Model

3.3 Roles and Services

The PrivateServer supports multiple, simultaneous operators. A database record entry is created by the PrivateServer for each operator and contains the operator name, authorization bits, quotas for operator temporary keys created by the operator, the certifier (CA) of the operator, and the minimum access level. The authorization mask controls the operator's permissions.

There are two primary roles an operator can hold, User/Application and Supervisor (Crypto-Officer):

3.3.1 Supervisor (Crypto-Officer) Role

The Supervisor is responsible for operator and key management, module initialization and startup, and the module's configuration. All authorization bits are turned on (i.e., FFFFFFFF) for the Supervisor, providing the following functionality:

- *Delete any key (besides Special-Purpose keys)*
- *Create users*
- *Retrieve user information*
- *Retrieve information about all open sessions*
- *Retrieve all information about any key, except its value.*

- *Revoke users*
- *Perform shutdown*
- *Perform firmware update*
- *Perform backup of all data in the module*
- *Restore previously backed up data*
- *Retrieve information from the log file*
- *Create a non-authenticated user*
- *Update user records*
- *Reset the log file*
- *Terminate a specific session*
- *Define Code mailing parameters*

In addition, the Supervisor can access all cryptographic and miscellaneous services, including:

- *Symmetric cryptographic services* - enable client applications use keys kept in PrivateServer for symmetric operations.
- *Asymmetric cryptographic services* – enable client applications use keys kept in PrivateServer for asymmetric operations.
- *Hashing services* – enable client applications use keys kept in PrivateServer for hashing operations.
- *All management services (keys, users, etc.)*
- *Administrative services*
- *Code Mailing services*

There can be only one individual holding the role of Supervisor. Only the Supervisor may possess the smartcards and passwords necessary to initialize and startup the module. This must be done locally, using the PrivateServer smartcard reader and a keyboard attached to the module. No other operator that can authenticate using this local interface. By connecting directly through the PrivateServer, the Supervisor has the ability to access certain management operations of the module, including:

- *Initializing the module and it's databases*
- *Starting the module*
- *Configuring the module's IP information*
- *Resetting a tamper condition*

3.3.2 User/Application Role

The User/Application is for accessing the cryptographic services provided by the module. The User logs into the module remotely through s device that communicates with the PrivateServer application program interface using the RSA challenge-response protocol. None of the authorization bits (see 3.3.1 for the functionality listing of those bits) are turned on for the User, the User can only access the following services:

- *Symmetric cryptographic services* - enable client applications use keys kept in PrivateServer for symmetric operations.

- *Asymmetric cryptographic services* – enable client applications use keys kept in PrivateServer for asymmetric operations.
- *Hashing services* – enable client applications use keys kept in PrivateServer for hashing operations.
- *All management services (keys, users, etc.)*
- *Administrative services*
- *Code mailing services*

A User must first authenticate to the module, there are two authentication schemes:

- RSA based Authentication scheme. In this scheme, after a successful authentication, an encrypted session is created. The RSA challenge-response protocol used by the module is a key distribution scheme, used to authenticate the operator and to establish a temporary session key (that is destroyed at the close of the session). Through this session, the operator may perform the cryptographic services for which they have permissions.
  The session keys (MAC and encryption/decryption) are negotiated during authentication of a user when creating a session. The PrivateServer creates these keys during the opening of an encrypted session, and they are destroyed when the session is terminated. These keys are temporary and are only stored in volatile memory.

- User ID / Password authentication scheme. In this scheme, after a successful authentication, a non-encrypted session is created.
  Through this session, the operator may perform the cryptographic services for which they have permissions.
  Any operation that either imports a key or exports a key from the module is restricted when this scheme is used. Also, any change password or set password operation must be done using the above RSA based secured session.

3.3.3 Authentication

The PrivateServer employs identity-based authentication of operators through either the RSA challenge-response mechanism or the User ID/Password authentication mechanism. The RSA challenge-response mechanism requires the exchange and verification of the operator's private key. All keys used for authentication are private keys generated externally and certified by a CA signature. The probability that random access will succeed is far less than one in 1,000,000 attempts using this authentication mechanism. In addition, the authentication provides 1 in 2^161/(1000*60) probability of a successful random attempt during a one-minute period.
In the case that the user ID/Password authentication scheme is used, the minimal password length is 6 bytes, this means that a random access will succeed is far less than one in 1,000,000 attempts using this authentication mechanism. In addition, upon a failed authentication attempt, a delay of 500ms will occur before the failure response is return to the client, since only up to 1000 sessions can be opened simultaneously 1000*120/(72^6) is far less than one in 100,000.
There is no limitation to the maximum password length.

The Supervisor possesses the smartcards and password necessary to initialize and startup the PrivateServer. The Supervisor can log into the module locally using the smartcards or remotely using the RSA challenge-response protocol. A Supervisor attempting to authenticate directly to the module through the keyboard port must use the startup smart card and password. The password must be at least 6 alphanumeric characters. This yields a minimum of 36^6 (over 1,000,000,000) possible combinations. Therefore the possibility of correctly guessing a password is less than 1 in 1,000,000. After fourteen failed authentication attempts the startup smartcard is locked and hence the possibility of randomly guessing a password in 60 seconds is less than 1 in 100,000. The module also suppresses feedback of authentication data being entered by returning '*' characters.

The maximum password length is 15 characters.

Remark: It is possible to configure PrivateServer such that there is no need to enter the startup smartcard as part of starting the module. In this configuration all Critical keys are loaded from the tamper device into the volatile memory upon startup.

3.3.4 Services

Table 3 provides a high-level summary of the services provided by the module.

Service	Information Summary
Key management and control	Secure storage and management of cryptographic keys (Triple-DES/AES keys, RSA public and private keys, ECDSA public and private keys, HMAC secret data, Special-purpose keys). In the case of user ID/password authentication, keys cannot be imported or exported in clear format to/from the HSM. Also, change password and set password operations can be done only when using an authenticated and encrypted session.
Public-key database	Centralized storage and management of public keys (RSA public keys and ECDSA public keys).
Data encryption and decryption	Symmetric [Triple-DES, AES] and Asymmetric Cryptography.
Digital signatures	Generate and verify digital signatures (RSA and ECDSA). The RSA digital signature generation supports the following schemes: PKCS#1 v1.5, PSS and ANSI X9.31. Also, digital signature verification service is supported based on the above algorithms.
Data hashing and data integrity	Generate message digests [SHA-1, SHA-256, SHA-384 and SHA-512 (FIPS 180-2)], HMAC/Triple-DES-CMAC/AES-CMAC/AES-CCM.
User authentication	Two-way user authentication using the RSA challenge-response key distribution mechanism. A smartcard token can be used. Also, for local access by the Supervisor a smartcard can be used. A user ID /password authentication mechanism can also be used.
Logging, auditing, secure auditing, administration, and management	Administrative and management operations as well as logging and auditing. Audit log can be signed.
Code Mailing	An interface to a Network printer for the purpose of producing a Code mailer.
Internal real-time clock	Used for accurate time stamps.
Get PrivateServer Details	This anonymous API function *csv_get_csv_cert* replies with the PrivateServer Identity and the public key of PrivateServer.

Table 3 – Service and Description

Note: When in non-FIPS mode services can use non approved algorithms such as DES and MD5 for legacy purposes.

Table 4 shows each specific service and which role has access to it.

SERVICES	**ROLE**	**INPUT**	**OUTPUT**
Delete any key	CO	Key-ID	Success Code
Create user	CO	New User Information	Success Code
Retrieve user information	CO	User ID	User Information
Retrieve information about all open sessions	CO	None	Information about active sessions
Retrieve al information about any key, except its value	CO	Key ID	Key Information
Revoke user	CO	User ID	Success Code
Perform shutdown	CO	None	None
Perform firmware update	CO	Updated Firmware	Success Code
Perform backups	CO	None	Encrypted Backup File
Restore backups	CO	Encrypted Backup File	Success Code
Retrieve log file	CO	None	Log File
Update user records	CO	User ID	Success Code
Update Key record	CO	Key ID	Success Code
Reset the log file	CO	None	Success Code
Terminate a session	CO	Session ID	Success Code
Defining a Code Mailing properties	CO	Mailing Info	Success Code
Symmetric cryptography	CO/User	Key ID, Input Buffer	Key ID, Output Buffer
Asymmetric cryptography	CO/User	Key ID, Input Buffer	Key ID, Output Buffer
Hashing	CO/User	Key ID, Input Buffer	Key ID, Output Buffer

Table 4 – Role Access to each Service

3.4 Cryptographic Algorithms and Secure Key Management

The PrivateServer supports a variety of cryptographic algorithms, and implements these algorithms based on the cryptographic standards. It provides the following FIPS 140-2-approved algorithms:

Data Encryption

- Triple-DES (ANSI X9.52) in ECB and CBC modes – 128 bits, 192 bits; Cert. #1286
- AES (FIPS PUB 197) in ECB and CBC modes – 128 bits, 192 bits and 256 bits; Cert. #1983

Data Packet Integrity

- Triple-DES-MAC - 128 bits, 196 bits; Cert. #1286
- HMAC Cert. #1196
- AES-CMAC; Cert. #1983
- Triple-DES-CMAC;Cert. #1286
- AES CCM; Cert. #1983

Message Digest

- SHA1; Cert. #1738
- SHA256; Cert. #1738
- SHA384; Cert. #1738
- SHA512; Cert. #1738

Random Number Generator

- FIPS 186-2; Cert. #1042

RSA Key Generation

- ANSI X9.31; Cert. #1029

Digital Signature Generation Algorithms (RSA Based)

- PKCS#1 v1.5; Cert. #1029
- PSS; Cert. #1029
- ANSI X9.31; Cert. #1029

Digital Signature Generation Algorithms (ECDSA Based)

- ECDSA; Cert. #288

Digital Signature Verification Algorithms (RSA Based)

- PKCS#1 v1.5; Cert. #1029
- PSS; Cert. #1029
- ANSI X9.31; Cert. #1029

Digital Signature Verification Algorithms (ECDSA Based)

- ECDSA Cert. #288

Non Approved Algorithms:

The module supports the following algorithms that are allowed in FIPS mode for key agreement and key establishment:

- RSA (key wrapping; key establishment methodology provides 80 bits of encryption strength).

The following algorithms cannot be used when module is operated in FIPS mode

Message Digest

- MD5
- ARDFP (an Algorithmic Research proprietary hashing algorithm)

Digital Signature Generation Algorithms

- RSA cipher only with ISO9796 padding

Data Encryption

- DES Stream (non-compliant)
- DES (FIPS 46-3) in ECB and CBC modes – 64 bits

Data Integrity

- DES-MAC (FIPS 113) – 64 bits

The PrivateServer stores all non-volatile keys in the database. The database is stored encrypted (with AES-128) on the PrivateServer's internal hard drive. Within the database, keys have properties associated with them. These properties determine which operations may be performed on a particular key and establish which users are authorized to carry out these operations.

There are two levels of access to the keys stored on the module, Owner and User. Each key maintains a list of owner IDs and User IDs. This should not be confused with the User Role as both levels of access are applicable to the User or Supervisor Role. The Owner of a key can perform all operations on the key and can grant or revoke key access rights to other entities. The User of a key may access it for cryptographic operations only and is not able to read the key or perform administrative functions on it.

When a user is authenticated to the PrivateServer, his/her user identity is defined. When accessing a key for the purpose of management or usage, this user identity is checked against the owner IDs list or user IDs list depending on the required operation.

User Keys are keys that are generated upon request or inputted by the user for various key operations. User keys consist of two types of keys: User Normal Keys, and User Temporary keys. Users choose what type of key they want to create or input. Users can generate or input any of the following key types: TDES 128 and 192 bit keys, AES 128, 192 and 256 bit keys, RSA 1024, 1536, 2048, 3072 and 4096 bit keys, ECDSA P-256, P-384 and P-521 curves. The only difference is that a User Normal key can be reused whereas a User Temporary key cannot. User Normal keys are stored in memory and then written to the database before the close of a user session. User Normal keys can be reloaded by the user for a new user session. User Temporary keys are only stored in memory and are erased upon close of a user session.

3.4.1 Initial Configuration

The PrivateServer has three AES-128 Critical keys which are generated using an external smart card reader. One half of each of the Critical keys is stored on the Startup smartcard and the other three halves are stored on the Initialization smart card. The Critical keys are then created by XORing the split keys from the Startup smart card and Initialization smart card and loading the result into the PrivateServer's volatile memory during startup. These keys are only stored in volatile memory. All three keys are erased from memory when the module is terminated.

The three critical keys are used for the following internal operations:

- Encrypting key values in the keys database
- Encrypting the database during a backup operation
- Checking the integrity of database records using a MAC key.

The Special-Purpose keys are only used for internal operations on the PrivateServer. These keys include the customer's organization-wide root public key, PrivateServer's RSA private/public key pair, the PrivateServer critical Keys, and the PrivateServer key for continuous operations context encryption.

Public keys and certificates stored in the public key database are inaccessible through the anonymous services (anonymous services are enabled when operating in non-FIPS 140-2 compliant mode). Certificates loaded onto the module must be signed by the organization's private key and this signature is verified before addition to the public key database.

In the FIPS 140-2 compliant mode of operation, all operator sessions are authenticated and encrypted so that no secret or private keys are passed in or out of the module unprotected.
In the case of a User ID/Password authenticated session, no key can be imported or export from the module in non-encrypted format. Also any change password or set password operations can be done over an authenticated and encrypted session.
The module also provides the ability to back up the key database in encrypted form.

Table 5 provides a list of all the keys, their key types, and access control.

Cryptographic Keys and CSPs	**Key Type**	**Key Generation/Establishment**	**ACCESS (R/W/X)**
Firmware Key	TDES 128 bit key, FIPS 46-2 (Some Firmware is encrypted by Algorithmic Research LTD this key is used to decrypt)	Hard Coded in the firmware	X
Critical Key for key value encryption of database keys	AES 128	Imported to PrivateServer during Module Initialization from Init Smartcard and Startup Smartcard	X
Critical Key for Backup encryption	AES 128	Imported to PrivateServer during Module Initialization from Init Smartcard and Startup Smartcard	X
Critical Key for database Record MAC calculation	AES 128	Imported to PrivateServer during Module Initialization from Init Smartcard and Startup Smartcard	X
Key for continuous operations context encryption	AES 128	Imported to PrivateServer during Module Initialization from Init Smartcard and Startup Smartcard	X
PrivateServer RSA Public/private key pair	RSA 1024 bit key	Generated during Module initialization	X
Organization Root Public Key	RSA 1024 bit key	Generated during Module initialization	X
Algorithmic Research/ AR RSA public key	RSA 1024 bit key (used to verify firmware signature)	Hard Coded in the firmware	X
Session encryption/decryption keys	AES 128	Temporary key is generated during session initiation	X
Session MAC keys	AES 128	Temporary key is generated during session initiation	X
User keys – (Two types: user Normal keys and user Temporary keys.)	Multiple key types: TDES 128 and 192 bit keys, AES 128, 192 and 256 bit keys, RSA Private Key 1024, 1536, 2048, 3072 and 4096 bit keys, RSA Public Key 1024, 1536, 2048, 3072 and 4096 bit keys, HMAC/Triple-DES-CMAC/AES-CMAC/AES-CCM secret data,	Generated by PrivateServer or imported from the PrivateServer client API (uploaded securely via PrivateServer secure connection).	X,R,W

	ECDSA P-256, P-384 and P-521 keys		
Public key certificates	RSA 1024 bit public keys stored in certificates	Uploaded in first connection attempt or the user	X,W
RSA based User Authentication	Authentication of operators uses RSA challenge-response mechanism. Authentication provides 1 in 2^161/(1000*60) probability of a successful random attempt during a one-minute period.	Used by user's key medium during session initiation	X
UID/Password Authentication	At least 6 character long. Yields a minimum of 72^6 (over 1,000,000,0000) possible combinations.	Used during session initiation	X
Password Authentication for console based authentication	At least 6 alphanumeric characters long. Yields a minimum of 36^6 (over 1,000,000,0000) possible combinations.	Used when operator access startup smartcard when physically accessing the module.	X
RNG seed	256 bits	Internally generated by PRNG	X
RNG seed key	128 bits	Internally generated by PRNG	X

[1.] X means that the key is used for executing cryptographic operation

[2.] W or R is relevant to User keys that can be imported or exported via encrypted session, depending on the key definitions.

Table 5 – Keys, Key Types, and Access

3.5 *Self Testing*

The PrivateServer monitors firmware operations through a set of self-tests to ensure proper operation in accordance with FIPS 140-2. The module includes the following self-tests:

Power-Up Self Tests:

Low-Level Hardware Tests: When power is first applied to the module, the hardware performs a series of checks to ensure it is functioning properly.

Firmware Integrity Test: After the hardware tests, the module performs RSA digital signature verification to ensure firmware has not been modified.

Cryptographic Algorithm KATs: Known Answer Tests (KATs) are run at power-up for the Triple DES and AES encryption/decryption, Message Authentication Codes and Hash Algorithms.

Triple-DES-CBC and Triple-DES-ECB KAT
AES128, AES192, AES256 CBC and ECB KAT
Triple-DES-MAC KAT
SHA-1 KAT
SHA-256 KAT
SHA-384 KAT
SHA-512 KAT
HMAC KAT
Triple-DES-CMAC KAT
AES-CMAC KAT
AES-CCM KAT
RNG KAT
RSA KAT

ECDSA Pairwise Consistency Test: All ECDSA operations are tested to ensure the correct operation of the ECDSA key generation and signatures.

Conditional Tests:

RSA Pairwise Consistency Test: All RSA operations are tested to ensure the correct operation of the RSA key generation, encryption/decryption, and signatures.
ECDSA Pairwise Consistency Test: All ECDSA operations are tested to ensure the correct operation of the ECDSA key generation, signatures.
Continuous RNG Test: PrivateServer random is based on a non-deterministic seed that is generated by an internal SmartCard inside the tamper device and a PRNG algorithm. The seed is updated every minute, checked for statistical errors and CRNG test is performed on it.
The output of the PRNG algorithm which is based on above seed is also checked for statistical errors CRNG test is performed on it.
If any of the tests fails, the module enters the error state.
Firmware load Test: Module firmware can only be remotely upgraded from the management system with proper authentication to the module. However, in order to strictly control the loading of new firmware to the PrivateServer, the new firmware must be digitally signed by Algorithmic Research. The load of a firmware update takes place using RSA signatures. The successful load of this update would render the module non FIPS validated unless the update has also been validated**.**

3.6 Mitigation of Other Attacks:

The PrivateServer does not include any mechanisms to prevent against special attacks.

4 FIPS 140-2 Level 3 Approved Mode

In FIPS approved mode of operation an authenticated session must be used. The supervisor role can define for every user whether to use the RSA based authentication scheme or the User ID/Password authentication scheme.

The supervisor role must use the two smartcards (Init Smartcard and Startup Smartcard) during pre-operational initialization of the module. The module is then configured to require the Startup Smartcard during standard initialization of the module. It is possible to configure PrivateServer such that there is no need to enter the Startup Smartcard during the initialization of the module. In this configuration all PrivateServer Critical keys content is kept inside the internal tamper device and erased upon a tamper event. This can be configured by using the Unattended Mode option in the PrivateServer's console. When configuring the module to require no Startup Smartcard, the supervisor will be prompted to insert the Startup Smartcard as well as enter the Startup Smartcard password. Upon success, this will enable the unattended startup configuration. The Unattended Mode option in the PrivateServer's console can also be disabled following the same procedures.

The module is shipped with either a FIPS 140-2 approved or non-approved mode. This is as requested by the customer at the time of purchase.

In order to switch a module to a FIPS 140-2 approved mode, set the *FIPS Mode* to *On*, in the PrivateServer settings dialog box. Once this configuration is accepted, the module is shutdown and restarts using that configuration file.

These instructions can be used to put the module back into the FIPS approved mode of operations if the module is placed into an non approved mode of operations.

When operating in an approved mode, certain functionality is unavailable. The anonymous functions, non-FIPS 140-2 compliant PrivateServer certificates, and non-FIPS 140-2 compliant challenge-response mechanism are all disabled.

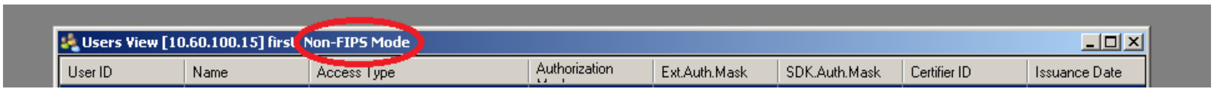
For FIPS 140-2 compliance, the session type for both Users and the Supervisor must be set to 3 (i.e., ACC_AUTHEN - authenticated and encrypted session) as depicted in Table 6. This can be set using the PrivateServer management utility. The CO's authorization mask is FFFFFFFF, and the User's authorization mask is 00000000. These can be set using the Algorithmic Research management utility provided or API calls.

Session Role	Non-Authenticated Session	Encrypted And Authenticated Session
User/ Application	No	Yes
Supervisor (Crypto-Officer)	No	Yes

Table 6 - Roles vs. Session Type

When is FIPS mode Cryptographic services shall only use FIPS 140-2-approved or allowed algorithms. A list of these algorithms can be found in section 2.4. The module also supports RSA key wrapping in FIPS mode for key establishment and key agreement.

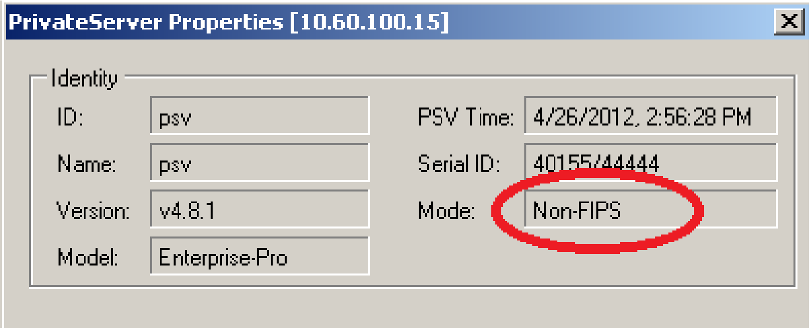
The FIPS mode is displayed in the title either the Users view, Keys view or Sessions views when using the PrivateServer Management utility. It is also displayed in the server->properties dialog of the PrivateServer management utility.



Users View [10.60.100.15] first Non-FIPS Mode

User ID	Name	Access Type	Authorization	Ext.Auth.Mask	SDK.Auth.Mask	Certifier ID	Issuance Date

Figure 1. Managment Utility FIPS mode indicator



PrivateServer Properties [10.60.100.15]

Identity

ID: psv

Name: psv

Version: v4.8.1

Model: Enterprise-Pro

PSV Time: 4/26/2012, 2:56:28 PM

Serial ID: 40155/44444

Mode: Non-FIPS

Figure 2. Properties dialog FIPS mode indicator

4.1 Module Inspection:

The cryptographic officer must perform a scheduled inspection of the module to detect tamper evidence. The cryptographic officer shall inspect three areas for tamper evidence:

1. The cryptographic officer shall inspect both of the Tamper Evident cans, which are located on the back of the module.
2. The cryptographic officer shall check the module’s front physical interfaces that are located behind the module’s front door.
3. The cryptographic officer shall remove the front ventilation cover to check for tamper evidence behind it.