SEMI E132-0310[E]
SPECIFICATION FOR EQUIPMENT CLIENT AUTHENTICATION AND AUTHORIZATION

This standard was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits & Reviews Subcommittee on December 16, 2009. Initially available at www.semi.org in February 2010. Originally published March 2004; previously published November 2008.

[E] This standard was editorially modified in June 2010 to correct errors in the related schema files. Changes were made to the SEMI E132.1 XML WSDL files.

1 Purpose

1.1 This specification describes a method for restricting access to communication with equipment by requiring clients to authenticate to the equipment before proceeding with subsequent communication, and provides a flexible authorization scheme to control client application access. Authorization allows parties who want to exchange information with or control the equipment to do so on a need-to-know and/or need-to-use basis.

1.2 The authorization scheme specified in this standard allows equipment vendor the flexibility to provide access control at any level of granularity, ranging from no access control restrictions, predefined role-based access control, to very fine-grained control.

2 Scope

2.1 This is a standard that applies to all semiconductor manufacturing equipment that requires authentication and authorization for its services. It does not apply to communication that is governed by the SEMI E30 communication and control state models.

2.2 This standard does not require data transmitted over an established session to be encrypted, encryption is only required as specified by the authentication protocol. It is assumed that the interface specified by this standard will be operating in an environment where there are no malicious attacks such as inside a closed factory network.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 An adjunct standard will need to be defined that maps specific technologies and protocols to the defined concepts in this document.

4 Referenced Standards and Documents

4.1 *SEMI Standards*

SEMI E30 — Generic Model for Communications and Control of SEMI Equipment (GEM)

SEMI E128 — Specification for XML Message Structures

4.2 *Other Standards*

Unified Modeling Language (UML) Specification, Version 1.4, OMG Specification 01-09-67; http://www.omg.org/technology/documents/modeling_spec_catalog.htm

Uniform Resource Identifiers (URI): Generic Syntax, IETF RFC 2396, August 1998; http://www.ietf.org/rfc/rfc2396.txt

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

SUPERSEDED

Documents Document Sharing Hub

Full document details: SEMI E132-0310E

5 Terminology

5.1 *Definitions*

5.1.1 *administration* — the process of configuring equipment client identities, associated credentials, and setting authorizations.

5.1.2 *Asymmetric Key Cryptography* — see Public Key Cryptography.

5.1.3 *authenticated communications* — communications with equipment compliant to this specification to establish an authenticated session, to grant authorizations and for administration. This includes any communications over the authenticated session. Note that authenticated communications are not necessarily encrypted.

5.1.4 *authenticated session* — also referred to as *session* in this document, this is an application-level concept defined by this specification and is independent of any physical or logical connection or communication protocol "session" layer used in the actual exchange of messages. An *authenticated session* or *session* follows the session establishment procedure, state models, and the defined services as described in this specification.

5.1.5 *authentication* — the process of determining whether a user or process is, in fact, who or what it is declared to be. In private and public computer networks, authentication is often achieved through the use of passwords. Knowledge of the password is assumed to guarantee that the user is authentic. This standard refers only to the authentication of application processes, and does not support or address authentication of human users.

5.1.6 *authorization* — the process of granting the privilege to perform a specific action to a user or process. Note that the user or process must be authenticated before authorization may take place.

5.1.7 *client* — an application process that communicates with the equipment to access equipment resources. This specification uses the terms "client" and "client application" interchangeably.

5.1.8 *communication session* — a series of two-way message exchanges between a client and the equipment. A session typically begins and ends at the request of the client.

5.1.9 *cryptographic algorithm* — a transformation of meaningful data into some meaningless data, a process called encryption. If the transformation is reversible, then the reverse process is called decryption. Typically it is computationally difficult to correctly decrypt the data without knowing both the algorithm used and some additional information, referred to as keys.

5.1.10 *encryption* — see cryptographic algorithm.

5.1.11 *message digest* — see one-way hash.

5.1.12 *one-way hash* — a one-way transformation of an arbitrary length of data into a fixed-length code. The transformation is computationally difficult to reverse, and unlikely to have collisions. The same message will always result in the same hash, and any slight modification to the original message will result in a different hash, thus providing message integrity. The terms One-way Hash, Hash and Digest are used interchangeably in this document.

5.1.13 *Persistent Session* — an *Authenticated Session* established that survives equipment shutdowns or power offs. Behavior of a persistent session during equipment shutdown and startup is as defined by the session state model described in this document.

5.1.14 *Public Key Cryptography (PKC)* — also known as Asymmetric Key Cryptography, a cryptographic algorithm that employs two keys for encryption and decryption of data. One key is designated as the public key and can be freely advertised, while the other key, designated as the private key, is never revealed. Data encrypted with one key can only be decrypted with the other key. PKC is widely used for key exchange and message signing.

5.1.15 *Secret Key Cryptography (SKC)* — also known as Symmetric Key Cryptography, a cryptographic algorithm that employs one single key for both encryption and decryption. SKC is widely used for bulk encryption of data due to its speed over PKC.

5.1.16 *Session* — in this document, the term "session" refers to an *Authenticated Session* unless specified otherwise. It has no relation to the sessions that may be defined elsewhere.

5.1.17 *Symmetric Key Cryptography* — see Secret Key Cryptography.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132-0310E

6 Conventions

6.1 *Unified Modeling Language (UML) Notation* — All class and sequence diagrams in this standard make use of the Unified Modeling Language notation.

6.2 *Attribute Tables* — The table below provides an example of the tables used to list and describe attributes of classes defined in this specification.

Table 1 Attribute Table Format

Attribute Name	*Definition*	*Form*
		See list below.

6.2.1 *Form* — Defines the data type of the attribute. The terms used to describe data types used in this column are defined in the SEMI Compilation of Terms, or are included as part of the specification. Refer to the compilation of terms for the definition of SEMI type name meanings.

6.2.2 *Association Tables* — The table below provides an example of the tables used to list and describe associations between classes defined in this specification.

Table 2 Association Table Format

Association Role Name	*Definition*	*Comments*

6.2.2.1 *Association Role Name* — The name of the association role being specified.

6.2.2.2 *Definition* — Describes the function or purpose of the association.

6.2.2.3 *Comments* — Any additional comments or notes regarding the association.

6.3 *Operation Definition Tables* — The table below provides an example of the tables used to list and describe the interface operations defined in this specification.

Table 3 Operation Definition

Operation	*Description*	*Type*	*Requestor/Sender*	*Responder/Receiver*
		See list below.		

6.3.1 *Operation* — Specifies the name of the operation.

6.3.2 *Type* — Specifies the messaging semantics of the operation. Only Request-Response (RR) and Fire-and-Forget (FF) semantics are used in this specification.

6.3.3 *Requestor/Sender* — For RR semantics, identifies the entity that makes the request, for FF semantics, identifies the entity that sends the message. Can be ‘Client’, ‘Admin’, or ‘Equipment’. Client can be any of the applications communicating to equipment compliant to this specification.

6.3.4 *Responder/Receiver* — For RR semantics, identifies the entity that responds to the request, for FF semantics, identifies the entity that receives the message. Can be ‘Client’, ‘Admin’, or ‘Equipment’.

6.4 *Operation Argument Definition Table* — The table below provides an example of the tables used to list and describe arguments for interface operations defined in this specification.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132-0310E

Table 4 Operation Argument Definitions

Argument	*Description*	*Kind*	*Form*
		See list below.	See list below.

6.4.1 *Argument* — Specifies the name of the argument.

6.4.2 *Kind* — Specifies whether the argument is an 'in', 'out', or 'error' argument for the operation. 'error' arguments always function as 'out' arguments, but indicate that the operation did not complete successfully. Not all errors possible for a specific operation will be listed, only select typical ones are.

6.4.3 *Form* — Defines the data type of the argument. The terms used to describe data types in this column are defined in the SEMI Compilation of Terms, or are included as part of the specification. Refer to the compilation of terms for the definition of SEMI type name meanings.

7 Background

7.1 In conventional semiconductor equipment communication, such as that defined by SEMI E30, it is assumed that only a single client application can gain access to the equipment for communication. This assumption was valid when the physical transport medium was based on a serial cable connection to the equipment. In such an environment, it is possible to ensure that only a single client application is permitted to interface with the equipment by only providing the serial connection to the machine on which the application runs.

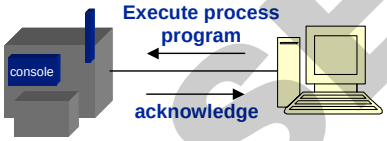

Figure 1
Equipment Communication via Serial Cable

7.2 As equipment communication moves from a serial cable medium to Ethernet, it becomes possible for any client on the network to send requests to the equipment, provided there is a path from the client node to the equipment. Without taking specific measures to secure communication with the equipment, it is not possible to ensure that only authorized clients are permitted to communicate with the equipment.

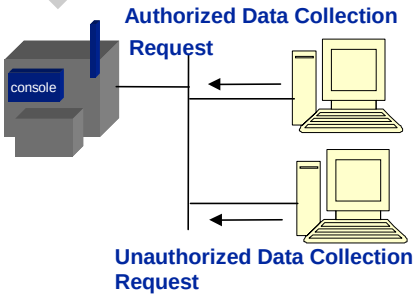

Figure 2
Equipment Communication via Ethernet

7.3 Even if the equipment accepts only a single connection request from the network, it is still vulnerable to a race condition among any clients that attempt to connect. This results in a "first-come, first-served" policy in which the first client to attempt a connection with the equipment is granted access, even if that client is not authorized to communicate with that specific equipment. Such an environment is insufficient for providing adequate protection of equipment resources.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132-0310E

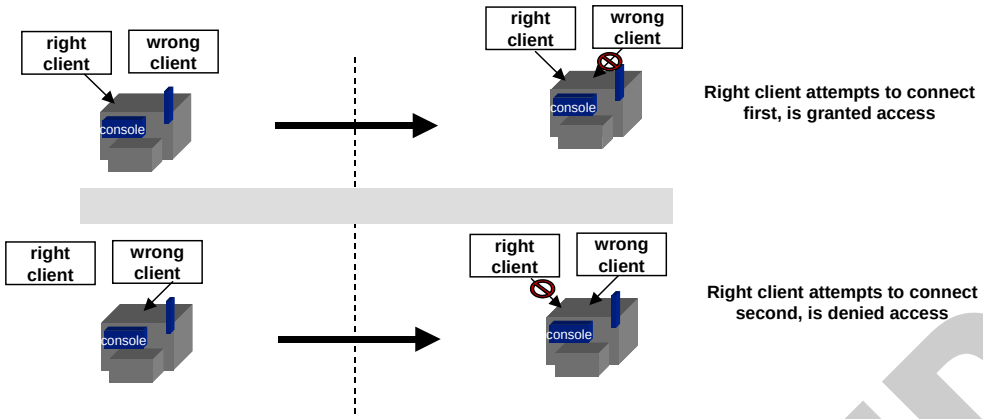

Figure 3
First-come First-served Communication

7.4 A more secure technique to ensure that only authorized clients are allowed to utilize equipment resources is to require clients to first prove their identity to the equipment. If the equipment knows a client's identity, then the equipment may be able to assume that the client is authorized to proceed with further requests.

8 Overview

8.1 This standard defines a mechanism for clients of manufacturing equipment to authenticate to the equipment prior to any subsequent communication. To facilitate this, both the authentication procedure itself and a means for establishing client identities with the equipment beforehand must be defined. This standard also defines a flexible scheme for authorization and mechanisms to support a centralized administration model.

8.2 *Central Administration Model* — A conceptual view of a central security administration model is presented in Figure 4 below.

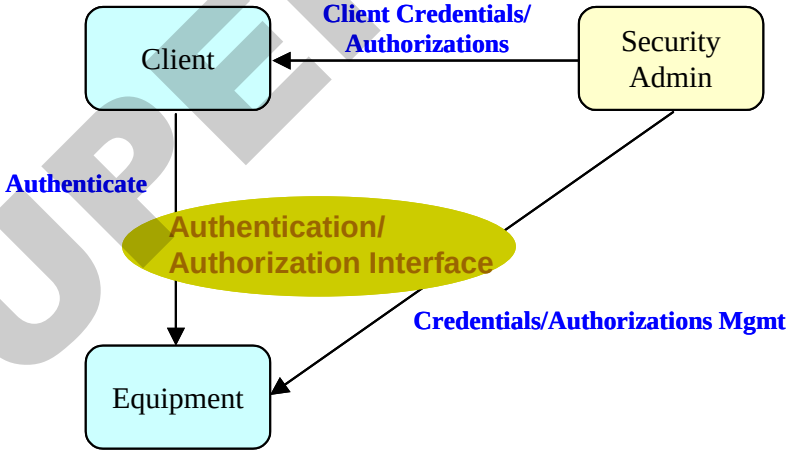

Figure 4
Security Administration Model

8.2.1 *Security Admin* — The security admin issues client credentials, establishes client authorizations and possibly client credentials on the equipment, defines custom roles and their authorizations. Client credentials are identifying information for the client. The security admin manages equipment authorizations and client credentials using an administration interface defined in this specification. The interface between security admin and client to communicate the credentials will not be defined in this document, and may be as simple as configurations inside the client application.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132-0310E

8.2.2 *Client* — The client holds any issued credentials assigned to it, and authenticates to the equipment to establish an authenticated session.

8.2.3 *Equipment* — The role of equipment in the administration model is to establish the security admin entity for the equipment, hold any defined authorizations, authenticate clients to establish a session, determine client authorization levels and enforce them for the established session.

8.2.4 *Interfaces* — The authentication, administration, and session establishment interfaces will be defined in later sections along with the concepts of establishing client identities and authorizations, though the actual technology and protocols used maybe defined in further detail in a separate adjunct standard.

8.3 *Challenge/Response Authentication*

8.3.1 The authentication process begins when a client first attempts to establish a communication session with the equipment. Upon receiving this request, the equipment challenges the client to prove its identity to the equipment. The client must encrypt some session-generated data using a private or secret key (e.g., a password) that only the claimed client would know. The equipment can verify client's claim using the corresponding public key or shared secret key assigned to the client to decrypt the data, if decrypted data matches plaintext then client is verified.

8.3.2 For communication technologies in which a separate outgoing connection must be established in order for the equipment to send events, the equipment must also verify that the receiving address is valid. This is achieved by sending a ping message to the client endpoint (see ¶ 14.1.1). This specification currently does not define a mechanism by which a client can transfer an established session to a different endpoint.

8.4 *Authorization*

8.4.1 Any client application wishing to invoke privileged services on the equipment must be authorized to do so. Authorization is implemented by granting access control privileges beforehand to defined clients and later checked against the client's requests. The general operational flow is as follows:

- *Authentication and Credentials Exchange* — Client application authenticates to equipment as to its identity and establishes an authenticated session.
- *Session Established* — Equipment optionally caches client application's privileges and associates these privileges with the newly-created session.
- *Service Request* — Client application makes a service request.
- *Privilege Validation* — Equipment validates the client application's request against its assigned privileges.
- *Session Closed* — Client application closes session, all session state information are reset or cleaned.

8.4.2 Since the client does not authenticate itself for each service request, privilege validation is performed against the authenticated session created for that client. This does not necessarily mean that the implementation of a session requires the use of a stateful or connection-oriented communication protocol, or that authentication must be performed during every protocol-level reconnect. If the implementation protocol is connectionless, the session can be identified by some mutually agreed upon session identifier, negotiated during the authentication process required to establish a session.

NOTE 1: This specification does not define a mechanism for controlling end-user access to equipment services or information. The authentication and authorization scheme defined in this specification applies at the application level only. Management of end-user access to equipment functions and information is strictly the responsibility of applications that make use of the authenticated communication sessions defined in this specification.

9 Authentication

9.1 *Authentication Message Flow* — The authentication message flow during a normal session establishment is presented in Figure 5. Arrows represent the major messages exchanged during the handshake, with side notes clarifying the key data exchanged in each message. Depending on the actual messaging/transport protocols and encryption algorithms used, some additional messages or data parameters may be added, but the concept remains the same. Specifics of algorithms and protocols will be detailed in a separate technology mapping specification. The terms server and equipment are used interchangeably.

9.1.1 *Session Request* — This message requests the equipment to initiate establishment of an authenticated session. Upon receiving this message, the equipment begins the authentication handshake as shown in Figure 5. If the request

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132-0310E

is not accepted, the equipment responds with a protocol specific alert or error message and terminates the connection, see Equipment Challenge.

9.1.2 *Equipment Challenge* — After the session request is accepted, the equipment responds with a challenge. The equipment selects the algorithms to be used for authentication (if applicable), provides its own identification and challenges the client to authenticate itself. The challenge can be a simple request to client to prove its identity. Note that errors can be returned instead of the normal parameters. Additional errors may be needed as required by the implementation protocol used to support the concepts.

9.1.3 *Client Authenticate* — The client responds to the server challenge, confirming the algorithms selected (if applicable), equipment's identification and also presents proof of client's own identity. The proof is typically a hash of some random data encrypted with a key that only the client would know and that the server can verify. A secret unique to the authentication protocol may also be generated by the client and sent to equipment in the response. Actual algorithms used for exchange of this secret will be defined by the specific authentication protocol specified in the adjunct standard.

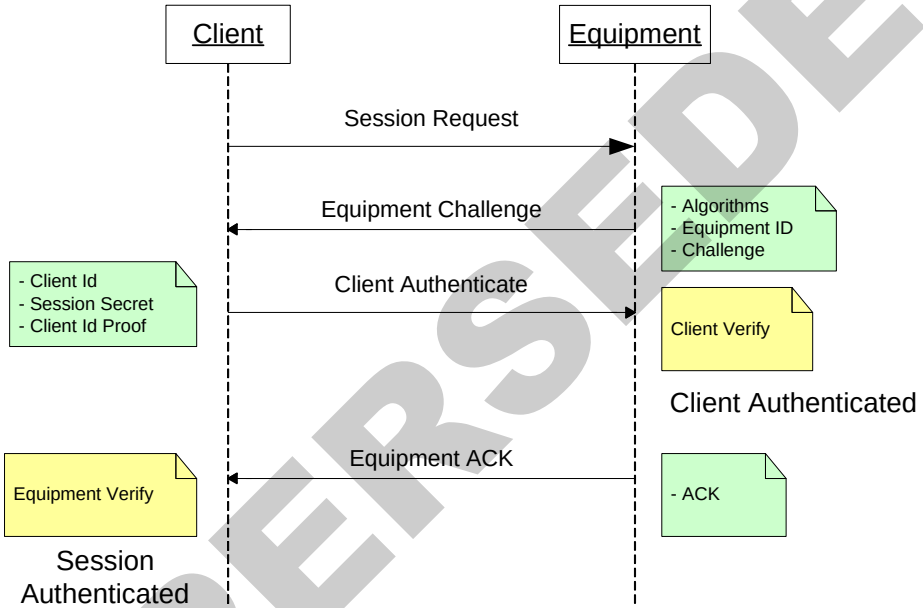

Figure 5
Authentication Message Flow

Table 5 Equipment Challenge Parameter Definitions

Parameter	*Description*	*Kind*	*Form*
Algorithms	Identifies the cryptographic algorithms to be used for key exchange, one-way hash functions and data encryption.	OUT	Text, with delimited fields to identify the algorithms used.
Equipment Id	Equipment identity, exact format as defined by the protocol chosen, and can be as simple as a name identifier such as in a password-based scheme.	OUT	Text.
Challenge	Request for client to authenticate itself, contains random data for client to sign, preventing replay.	OUT	Text.
error	Illegal or unexpected message.	error	Defined by the specific authentication protocol used.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132-0310E

Table 6 Client Authenticate Parameter Definitions

Parameter	*Description*	*Kind*	*Form*
Client Id	Client identity, exact format as defined by the protocol chosen, and can be as simple as a name identifier such as in a password-based scheme.	IN	Text.
Session Secret	Randomly generated data unique to the authentication process. This is encrypted using the equipment's public key or an established shared secret key, depending on the actual protocol used and algorithm chosen for key exchange.	IN	Binary, encrypted.
Client Id Proof	Encryption on some random data using a secret that only this client would know, proving client's claim of identity. This serves as a signature. The random data can be the data from equipment's challenge or derived from it.	IN	Binary, encrypted.
error	Illegal or unexpected message.	error	Defined by the specific authentication protocol used.

9.1.4 *Equipment ACK* — Equipment verifies client's response and confirms the exchanged session secret in this final authentication protocol message. The acknowledgement contains a parameter that can be a hash of the secret or its derived secret. The derivation algorithm for derived session secret is known both to the equipment and the client, and is part of the protocol specification. This allows client to verify the exchange by verifying the hash. Subsequent communication over the session will make use of the derived secret keys. Note that this standard does not require data be encrypted over the established session, the derived secret is only used to identify the session and hence the authorizations of the authenticated client. Additional errors may be defined as required by the protocol chosen, and technology used by the implementations.

Table 7 Equipment ACK Argument Definitions

Parameter	*Description*	*Kind*	*Form*
ACK	Hash of data derived from the exchanged session secret, this proves equipment's correct decryption of the secret. Hash algorithms as agreed to at the beginning of handshake.	OUT	Text, encrypted.
error	Illegal or unexpected message received.	error	Defined by the specific authentication protocol used.
error	Illegal parameter.	error	Defined by the specific authentication protocol used.
error	Authentication failure.	error	Defined by the specific authentication protocol used.

9.2 *Session Identifier*

9.2.1 The session id returned from the EstablishSession request (see ¶ 12.2.5) can be used to identify the session for subsequent communications and to enforce authorizations. The identifier will be embedded in special headers in the message transport, and not as formal parameters in the communication messages thus preserving their original structure as defined by the respective interface specifications.

10 Authorization

10.1 *Authorization Overview*

10.1.1 Any client wishing to invoke privileged services on the equipment must be authorized to do so. Authorization is implemented by granting access control privileges beforehand to defined clients and later checked against the client's requests. The total set of available privileges may include standardized service-level access

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132-0310E

control, such as privileges to invoke specific services or functions, but may also include supplier-defined finer-grained privileges. A default privilege to allow access to all services available from an authenticated session is defined in this specification, and must be supported by all implementations. A user of the equipment therefore has the option to reduce security enforcement on the equipment to authentication alone, achieved by assigning this all-access privilege to all defined clients.

10.1.2 Figure 6 provides an abstract overview of the relationships among the Role, Principal, and Privilege concepts defined in this specification. A Principal is either assigned to one and only one Role, or to one or more Privileges. A Role is defined as a uniquely-identifiable set of 1 or more Privileges, and cannot be composed from other Roles. Zero or more Principals can be assigned to at most one Role. These relationships are enforced through the management of Access Control List entries, described in ¶ 10.2.

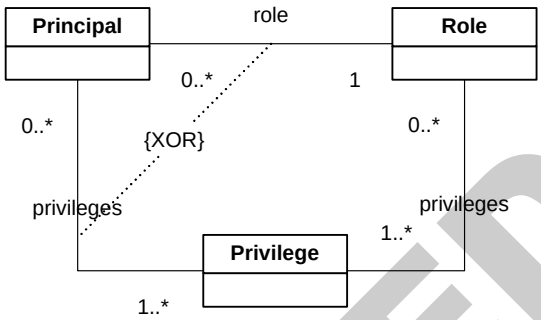

Figure 6
Abstract Model of ACL

10.2 *Access Control List*

10.2.1 Access control privileges are assigned to defined clients (principals) by using an Access Control List (ACL). An ACL is a collection of ACL entries, each entry granting a subset of the available privileges to a specific subject. A subject can represent a specific principal, a uniquely-named role, or all principals that successfully authenticate with the equipment. A principal can have privileges granted to it through either an explicit list of privileges, or indirectly through assignment to a role. The equipment shall maintain the ACL defined in this specification in non-volatile memory.

10.2.2 Roles are used to simplify the management of privilege assignments to principals. Each unique role has an ACL entry that specifies the privileges assigned to that role. Those privileges can then be granted to principals through a RoleAssignment ACL entry for each principal that should have those privileges. A principal must either have a RoleAssignment or an explicit list of privileges assigned to it (PrivilegeAssignment); it is not permitted to have both.

10.2.3 The equipment shall refuse session requests from clients that have no ACLEntry, either explicitly through a role or privilege assignment, or implicitly through the existence of an ACLEntry for all authenticated principals.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132-0310E