Network Working Group
Request for Comments: 5027
Updates: 3312
Category: Standards Track

F. Andreasen
D. Wing
Cisco Systems

October 2007

Security Preconditions for Session Description Protocol (SDP) Media Streams

Status of This Memo

Abstract

This document defines a new security precondition for the Session Description Protocol (SDP) precondition framework described in RFCs 3312 and 4032. A security precondition can be used to delay session establishment or modification until media stream security for a secure media stream has been negotiated successfully.

Table of Contents

1. Introduction

The concept of a Session Description Protocol (SDP) [RFC4566] precondition is defined in [RFC3312] as updated by [RFC4032]. A precondition is a condition that has to be satisfied for a given media stream in order for session establishment or modification to proceed. When a (mandatory) precondition is not met, session progress is delayed until the precondition is satisfied or the session establishment fails. For example, RFC 3312 defines the Quality-of-Service precondition, which is used to ensure availability of network resources prior to establishing (i.e., alerting) a call.

Media streams can either be provided in cleartext and with no integrity protection, or some kind of media security can be applied, e.g., confidentiality and/or message integrity. For example, the Audio/Video profile of the Real-Time Transfer Protocol (RTP) [RFC3551] is normally used without any security services whereas the Secure Real-time Transport Protocol (SRTP) [SRTP] is always used with security services. When media stream security is being negotiated, e.g., using the mechanism defined in SDP Security Descriptions [SDESC], both the offerer and the answerer [RFC3264] need to know the cryptographic parameters being used for the media stream; the offerer may provide multiple choices for the cryptographic parameters, or the cryptographic parameters selected by the answerer may differ from those of the offerer (e.g., the key used in one direction versus the other). In such cases, to avoid media clipping, the offerer needs to receive the answer prior to receiving any media packets from the answerer. This can be achieved by using a security precondition, which ensures the successful negotiation of media stream security parameters for a secure media stream prior to session establishment or modification.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Security Precondition Definition

The semantics for a security precondition are that the relevant cryptographic parameters (cipher, key, etc.) for a secure media stream are known to have been negotiated in the direction(s) required. If the security precondition is used with a non-secure media stream, the security precondition is by definition satisfied. A secure media stream is here defined as a media stream that uses some kind of security service (e.g., message integrity,

confidentiality, or both), regardless of the cryptographic strength of the mechanisms being used.

As an extreme example of this, Secure RTP (SRTP) using the NULL encryption algorithm and no message integrity would be considered a secure media stream whereas use of plain RTP would not. Note though, that Section 9.5 of [SRTP] discourages the use of SRTP without message integrity.

Security preconditions do not guarantee that an established media stream will be secure. They merely guarantee that the recipient of the media stream packets will be able to perform any relevant decryption and integrity checking on those media stream packets. Please refer to Section 5 for further security considerations.

The security precondition type is defined by the string "sec" and hence we modify the grammar found in RFC 3312 as follows:

```
precondition-type  =  "sec" / "qos" / token
```

RFC 3312 defines support for two kinds of status types, namely segmented and end-to-end. The security precondition-type defined here MUST be used with the end-to-end status type; use of the segmented status type is undefined.

A security precondition can use the strength-tag "mandatory", "optional", or "none".

When a security precondition with a strength-tag of "mandatory" is received in an offer, session establishment or modification MUST be delayed until the security precondition has been met, i.e., the relevant cryptographic parameters (cipher, key, etc.) for a secure media stream are known to have been negotiated in the direction(s) required. When a mandatory security precondition is offered, and the answerer cannot satisfy the security precondition (e.g., because the offer was for a secure media stream, but it did not include the necessary parameters to establish the secure media stream keying material for example), the offered media stream MUST be rejected as described in RFC 3312.

The delay of session establishment defined here implies that alerting of the called party MUST NOT occur and media for which security is being negotiated MUST NOT be exchanged until the precondition has been satisfied. In cases where secure media and other non-media data is multiplexed on a media stream (e.g., when Interactive Connectivity Establishment [ICE] is being used), the non-media data is allowed to be exchanged prior to the security precondition being satisfied.

When a security precondition with a strength-tag of "optional" is received in an offer, the answerer MUST generate its answer SDP as soon as possible. Since session progress is not delayed in this case, the answerer does not know when the offerer is able to process secure media stream packets and hence clipping may occur. If the answerer wants to avoid clipping and delay session progress until he knows the offerer has received the answer, the answerer MUST increase the strength of the security precondition by using a strength-tag of "mandatory" in the answer. Note that use of a mandatory precondition in an offer requires the presence of a SIP "Require" header field containing the option tag "precondition": Any SIP UA that does not support a mandatory precondition will consequently reject such requests (which also has unintended ramifications for SIP forking that are known as the Heterogeneous Error Response Forking Problem (see e.g., [HERFP]). To get around this, an optional security precondition and the SIP "Supported" header field containing the option tag "precondition" can be used instead.

When a security precondition with a strength-tag of "none" is received, processing continues as usual. The "none" strength-tag merely indicates that the offerer supports the following security precondition - the answerer MAY upgrade the strength-tag in the answer as described in [RFC3312].

The direction tags defined in RFC 3312 are interpreted as follows:

* send: Media stream security negotiation is at a stage where it is possible to send media packets to the other party and the other party will be able to process them correctly from a security point of view, i.e., decrypt and/or integrity check them as necessary. The definition of "media packets" includes all packets that make up the media stream. In the case of Secure RTP for example, it includes SRTP as well as SRTCP. When media and non-media packets are multiplexed on a given media stream (e.g., when ICE is being used), the requirement applies to the media packets only.

* recv: Media stream security negotiation is at a stage where it is possible to receive and correctly process media stream packets sent by the other party from a security point of view.

The precise criteria for determining when the other party is able to correctly process media stream packets from a security point of view depend on the secure media stream protocol being used as well as the mechanism by which the required cryptographic parameters are negotiated.

We here provide details for SRTP negotiated through SDP security descriptions as defined in [SDESC]:

* When the offerer requests the "send" security precondition, it needs to receive the answer before the security precondition is satisfied. The reason for this is twofold. First, the offerer needs to know where to send the media. Secondly, in the case where alternative cryptographic parameters are offered, the offerer needs to know which set was selected. The answerer does not know when the answer is actually received by the offerer (which in turn will satisfy the precondition), and hence the answerer needs to use the confirm-status attribute [RFC3312]. This will make the offerer generate a new offer showing the updated status of the precondition.

* When the offerer requests the "recv" security precondition, it also needs to receive the answer before the security precondition is satisfied. The reason for this is straightforward: The answer contains the cryptographic parameters that will be used by the answerer for sending media to the offerer; prior to receipt of these cryptographic parameters, the offerer is unable to authenticate or decrypt such media.

When security preconditions are used with the Key Management Extensions for the Session Description Protocol (SDP) [KMGMT], the details depend on the actual key management protocol being used.

After an initial offer/answer exchange in which the security precondition is requested, any subsequent offer/answer sequence for the purpose of updating the status of the precondition for a secure media stream SHOULD use the same key material as the initial offer/answer exchange. This means that the key-mgmt attribute lines [KMGMT], or crypto attribute lines [SDESC] in SDP offers, that are sent in response to SDP answers containing a confirm-status field [RFC3312] SHOULD repeat the same data as that sent in the previous SDP offer. If applicable to the key management protocol or SDP security description, the SDP answers to these SDP offers SHOULD repeat the same data in the key-mgmt attribute lines [KMGMT] or crypto attribute lines [SDESC] as that sent in the previous SDP answer.

Of course, this duplication of key exchange during precondition establishment is not to be interpreted as a replay attack. This issue may be solved if, e.g., the SDP implementation recognizes that the key management protocol data is identical in the second offer/answer exchange and avoids forwarding the information to the security layer for further processing.

Offers with security preconditions in re-INVITEs or UPDATEs follow the rules given in Section 6 of RFC 3312, i.e.:

> "Both user agents SHOULD continue using the old session parameters until all the mandatory preconditions are met. At that moment, the user agents can begin using the new session parameters."

At that moment, we furthermore require that user agents MUST start using the new session parameters for media packets being sent. The user agents SHOULD be prepared to process media packets received with either the old or the new session parameters for a short period of time to accommodate media packets in transit. Note that this may involve iterative security processing of the received media packets during that period of time. Section 8 in [RFC3264] lists several techniques to help alleviate the problem of determining when a received media packet was generated according to the old or new offer/answer exchange.

4. Examples

4.1. SDP Security Descriptions Example

The call flow of Figure 1 shows a basic session establishment using the Session Initiation Protocol [SIP] and SDP security descriptions [SDESC] with security descriptions for the secure media stream (SRTP in this case).

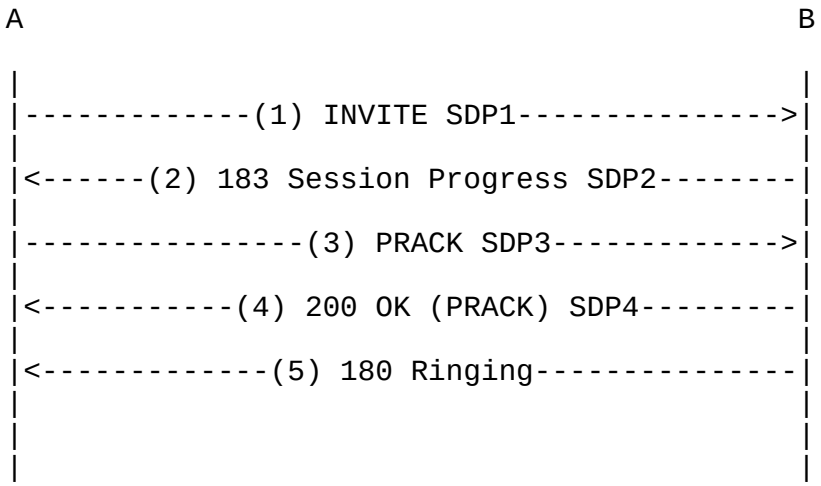

Figure 1: Security Preconditions with SDP Security Descriptions Example

The SDP descriptions of this example are shown below - we have omitted the details of the SDP security descriptions as well as any SIP details for clarity of the security precondition described here:

SDP1: A includes a mandatory end-to-end security precondition for both the send and receive direction in the initial offer as well as a "crypto" attribute (see [SDESC]), which includes keying material that can be used by A to generate media packets. Since B does not know any of the security parameters yet, the current status (see RFC 3312) is set to "none". A's local status table (see RFC 3312) for the security precondition is as follows:

Direction	Current	Desired Strength	Confirm
send	no	mandatory	no
recv	no	mandatory	no

and the resulting offer SDP is:

```
m=audio 20000 RTP/SAVP 0
c=IN IP4 192.0.2.1
a=curr:sec e2e none
a=des:sec mandatory e2e sendrecv
a=crypto:foo...
```

SDP2: When B receives the offer and generates an answer, B knows the (send and recv) security parameters of both A and B. From a security perspective, B is now able to receive media from A, so B's "recv" security precondition is "yes". However, A does not know any of B's SDP information, so B's "send" security precondition is "no". B's local status table therefore looks as follows:

Direction	Current	Desired Strength	Confirm
send	no	mandatory	no
recv	yes	mandatory	no

B requests A to confirm when A knows the security parameters used in the send and receive direction (it would suffice for B to ask for confirmation of A's send direction only) and hence the resulting answer SDP becomes:

```
m=audio 30000 RTP/SAVP 0
c=IN IP4 192.0.2.4
a=curr:sec e2e recv
a=des:sec mandatory e2e sendrecv
a=conf:sec e2e sendrecv
a=crypto:bar...
```

SDP3: When A receives the answer, A updates its local status table based on the rules in RFC 3312. A knows the security parameters of both the send and receive direction and hence A's local status table is updated as follows:

Direction	Current	Desired Strength	Confirm
send	yes	mandatory	yes
recv	yes	mandatory	yes

Since B requested confirmation of the send and recv security preconditions, and both are now satisfied, A immediately sends an updated offer (3) to B showing that the security preconditions are satisfied:

```
m=audio 20000 RTP/SAVP 0
c=IN IP4 192.0.2.1
a=curr:sec e2e sendrecv
a=des:sec mandatory e2e sendrecv
a=crypto:foo...
```

Note that we here use PRACK [RFC3262] instead of UPDATE [RFC3311] since the precondition is satisfied immediately, and the original offer/answer exchange is complete.

SDP4: Upon receiving the updated offer, B updates its local status table based on the rules in RFC 3312, which yields the following:

Direction	Current	Desired Strength	Confirm
send	yes	mandatory	no
recv	yes	mandatory	no

B responds with an answer (4) that contains the current status of the security precondition (i.e., sendrecv) from B's point of view:

```
m=audio 30000 RTP/SAVP 0
c=IN IP4 192.0.2.4
a=curr:sec e2e sendrecv
a=des:sec mandatory e2e sendrecv
a=crypto:bar...
```

B's local status table indicates that all mandatory preconditions have been satisfied, and hence session establishment resumes; B returns a 180 (Ringing) response (5) to indicate alerting.

4.2. Key Management Extension for SDP Example

The call flow of Figure 2 shows a basic session establishment using the Session Initiation Protocol [SIP] and Key Management Extensions for SDP [KMGMT] with security descriptions for the secure media stream (SRTP in this case):

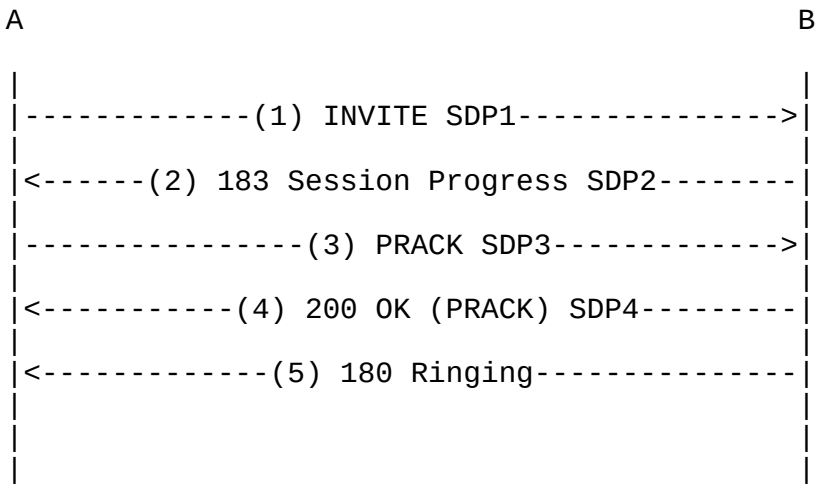

Figure 2: Security Preconditions with Key Management Extensions for SDP Example

The SDP descriptions of this example are shown below - we show an example use of MIKEY [MIKEY] with the Key Management Extensions, however we have omitted the details of the MIKEY parameters as well as any SIP details for clarity of the security precondition described here:

SDP1: A includes a mandatory end-to-end security precondition for both the send and receive direction in the initial offer as well as a "key-mgmt" attribute (see [KMGMT]), which includes keying material that can be used by A to generate media packets. Since B does not know any of the security parameters yet, the current status (see RFC 3312) is set to "none". A's local status table (see RFC 3312) for the security precondition is as follows:

Direction	Current	Desired Strength	Confirm
send	no	mandatory	no
recv	no	mandatory	no

and the resulting offer SDP is:

```
m=audio 20000 RTP/SAVP 0
c=IN IP4 192.0.2.1
a=curr:sec e2e none
a=des:sec mandatory e2e sendrecv
a=key-mgmt:mikey AQAFgM0X...
```

SDP2: When B receives the offer and generates an answer, B knows the (send and recv) security parameters of both A and B. B generates keying material for sending media to A, however, A does not know B's keying material, so the current status of B's "send" security precondition is "no". B does know A's SDP information, so B's "recv" security precondition is "yes". B's local status table therefore looks as follows:

Direction	Current	Desired Strength	Confirm
send	no	mandatory	no
recv	yes	mandatory	no

B requests A to confirm when A knows the security parameters used in the send and receive direction and hence the resulting answer SDP becomes:

```
m=audio 30000 RTP/SAVP 0
c=IN IP4 192.0.2.4
a=curr:sec e2e recv
a=des:sec mandatory e2e sendrecv
a=conf:sec e2e sendrecv
a=key-mgmt:mikey AQAFgM0X...
```

Note that the actual MIKEY data in the answer differs from that in the offer; however, we have only shown the initial and common part of the MIKEY value in the above.

SDP3: When A receives the answer, A updates its local status table based on the rules in RFC 3312. A now knows all the security parameters of both the send and receive direction and hence A's local status table is updated as follows:

Direction	Current	Desired Strength	Confirm
send	yes	mandatory	yes
recv	yes	mandatory	yes

Since B requested confirmation of the send and recv security preconditions, and both are now satisfied, A immediately sends an updated offer (3) to B showing that the security preconditions are satisfied:

```
m=audio 20000 RTP/SAVP 0
c=IN IP4 192.0.2.1
a=curr:sec e2e sendrecv
a=des:sec mandatory e2e sendrecv
a=key-mgmt:mikey AQAFgM0X...
```

SDP4: Upon receiving the updated offer, B updates its local status table based on the rules in RFC 3312, which yields the following:

Direction	Current	Desired Strength	Confirm
send	yes	mandatory	no
recv	yes	mandatory	no

B responds with an answer (4) that contains the current status of the security precondition (i.e., sendrecv) from B's point of view:

```
m=audio 30000 RTP/SAVP 0
c=IN IP4 192.0.2.4
a=curr:sec e2e sendrecv
a=des:sec mandatory e2e sendrecv
a=key-mgmt:mikey AQAFgM0X...
```

B's local status table indicates that all mandatory preconditions have been satisfied, and hence session establishment resumes; B returns a 180 (Ringing) response (5) to indicate alerting.

5. Security Considerations

In addition to the general security considerations for preconditions provided in RFC 3312, the following security issues should be considered.

Security preconditions delay session establishment until cryptographic parameters required to send and/or receive media for a media stream have been negotiated. Negotiation of such parameters can fail for a variety of reasons, including policy preventing use of certain cryptographic algorithms, keys, and other security parameters. If an attacker can remove security preconditions or downgrade the strength-tag from an offer/answer exchange, the attacker can thereby cause user alerting for a session that may have no functioning media. This is likely to cause inconvenience to both the offerer and the answerer. Similarly, security preconditions can

be used to prevent clipping due to race conditions between an offer/answer exchange and secure media stream packets based on that offer/answer exchange. If an attacker can remove or downgrade the strength-tag of security preconditions from an offer/answer exchange, the attacker can cause clipping to occur in the associated secure media stream.

Conversely, an attacker might add security preconditions to offers that do not contain them or increase their strength-tag. This in turn may lead to session failure (e.g., if the answerer does not support it), heterogeneous error response forking problems, or a delay in session establishment that was not desired.

Use of signaling integrity mechanisms can prevent all of the above problems. Where intermediaries on the signaling path (e.g., SIP proxies) are trusted, it is sufficient to use only hop-by-hop integrity protection of signaling, e.g., IPSec or TLS. In all other cases, end-to-end integrity protection of signaling (e.g., S/MIME) MUST be used. Note that the end-to-end integrity protection MUST cover not only the message body, which contains the security preconditions, but also the SIP "Supported" and "Require" headers, which may contain the "precondition" option tag. If only the message body were integrity protected, removal of the "precondition" option tag could lead to clipping (when a security precondition was otherwise to be used), whereas addition of the option tag could lead to session failure (if the other side does not support preconditions).

As specified in Section 3, security preconditions do not guarantee that an established media stream will be secure. They merely guarantee that the recipient of the media stream packets will be able to perform any relevant decryption and integrity checking on those media stream packets.

Current SDP [RFC4566] and associated offer/answer procedures [RFC3264] allows only a single type of transport protocol to be negotiated for a given media stream in an offer/answer exchange. Negotiation of alternative transport protocols (e.g., plain and secure RTP) is currently not defined. Thus, if the transport protocol offered (e.g., secure RTP) is not supported, the offered media stream will simply be rejected. There is however work in progress to address that. For example, the SDP Capability Negotiation framework [SDPCN] defines a method for negotiating the use of a secure or a non-secure transport protocol by use of SDP and the offer/answer model with various extensions.

Such a mechanism introduces a number of security considerations in general, however use of SDP Security Preconditions with such a

mechanism introduces the following security precondition specific security considerations:

A basic premise of negotiating secure and non-secure media streams as alternatives is that the offerer's security policy allows for non-secure media. If the offer were to include secure and non-secure media streams as alternative offers, and media for either alternative may be received prior to the answer, then the offerer may not know if the answerer accepted the secure alternative. An active attacker thus may be able to inject malicious media stream packets until the answer (indicating the chosen secure alternative) is received. From a security point of view, it is important to note that use of security preconditions (even with a mandatory strength-tag) would not address this vulnerability since security preconditions would effectively apply only to the secure media stream alternatives. If the non-secure media stream alternative was selected by the answerer, the security precondition would be satisfied by definition, the session could progress and (non-secure) media could be received prior to the answer being received.

6. IANA Considerations

IANA has registered an RFC 3312 precondition type called "sec" with the name "Security precondition". The reference for this precondition type is the current document.

7. Acknowledgements

The security precondition was defined in earlier versions of RFC 3312. RFC 3312 contains an extensive list of people who worked on those earlier versions, which are acknowledged here as well. The authors would additionally like to thank David Black, Mark Baugher, Gonzalo Camarillo, Paul Kyzivat, and Thomas Stach for their comments on this document.

8. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3312] Camarillo, G., Ed., Marshall, W., Ed., and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", RFC 3312, October 2002.

[RFC4032] Camarillo, G. and P. Kyzivat, "Update to the Session Initiation Protocol (SIP) Preconditions Framework", RFC 4032, March 2005.

[SIP] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

9. Informative References

[SDESC] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[SRTP] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[ICE] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Multimedia Session Establishment Protocols", Work in Progress, September 2007.

[KMGMT] Arkko, J., Lindholm, F., Naslund, M., Norrman, K., and E. Carrara, "Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)", RFC 4567, July 2006.

[MIKEY] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

[HERFP] Mahy, R., "A Solution to the Heterogeneous Error Response Forking Problem (HERFP) in the Session Initiation Protocol (SIP)", Work in Progress, March 2006.

[SDPCN] Andreasen, F., "SDP Capability Negotiation", Work in Progress, July 2007.

Authors' Addresses

Flemming Andreasen
Cisco Systems, Inc.
499 Thornall Street, 8th Floor
Edison, New Jersey 08837 USA

EMail: fandreas@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134 USA

EMail: dwing@cisco.com

Full Copyright Statement

Intellectual Property