Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2012

D. Zhang
Huawei Technologies Co.,Ltd
D. Kuptsov
HIIT
S. Shen
CNNIC

March 9, 2012

Host Identifier Revocation in HIP
draft-irtf-hiprg-revocation-05

Abstract

This document mainly analyzes the key revocation issue with host identifiers (HIs) in the Host Identity Protocol (HIP). Generally, key revocation is an important functionality of key management systems; it is concerned with the issues of removing cryptographic keys from operational use when they are not secure or not secure enough any more. This functionality is particularly important for the security systems expected to execute for long periods. This document also attempts to investigate several issues that a designer of HI revocation mechanisms need to carefully consider.

Requirements Language

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 RFC 2119 [RFC2119].

Table of Contents

1. Introduction

In HIP architecture [RFC5201] along with the ephemeral keys, derived during the protocol run between two peers, a HIP-enabled host is provided with a public/private key pair which are considered to be used for a long period of time. When two HIP hosts attempt to establish a connection (e.g., a TCP session) they use the public half of the key to represent it's identify (HI). In literature this phase is usually referred to as identification process. Usually, HIs being the public information can be communicated as a plaintext, unless one requires also support for privacy. On the other hand, hosts use the private halves of the keys to prove that they are the genuine holders of the corresponding HIs. This process is commonly defined as authentication. As the name implies the private halves should always be kept in a secret. Clearly, the security of many HIP deployments largely depends on the security of the private/public key pairs. If the private key pair of a HIP host is revealed (by accident or intentionally), an attacker can potentially impersonate the victim to carry out attacks without being detected for long period of time.

It has been widely recognized that a cryptographic key (which can be either a symmetric key or a public key) should have a reasonable valid period [Recommendations]. After being employed for a certain period, a cryptographic may become susceptible to cryptanalysis: As time elapses, an attacker can collect enough material (e.g., encrypted data, signatures and associated plain texts, etc.) to compromise the key. On the other extreme, the accidental key disclosures is yet another threat. For instance, such situation can occur as a result of improper key management policies or hardware compromise. It is therefore inevitable that the design of a security system, which is expected to be operational for a long period of time, will include the mechanisms for efficient and secure cryptographic keys management.

It is reasonable to assume that after HIP has been widely adopted lots of users may not have enough security knowledge to correctly deal with their insecure cryptographic keys, and thus an automatic key revocation solution will be desired. So far, only transient (session) key revocation issues have been discussed within the HIP framework. Briefly, HIP allows two communicating hosts to update their transient keys securely at run time. However, the key revocation issues with permanent keys (i.e., HIs) have not been well explored yet.

During the discussion of this draft, it is assumed that 1) an attacker cannot compromise an HI by brutal force during a reasonable long period but may need to be removed from usuage for certain reasons, 2) an attacker can intercept and modify the packets

transported between honest HIP hosts.

2. Terminology

BEX (Base Exchange): The handshaking protocol defined in [RFC5201], which enable two HIP hosts use their public keys pairs to generate key materials for subsequent communication.

HI (Host Identifier): A public key kept by a HIP host to represent the identity of the host.

HIT (Host Identity Tag): A 128-bit value generated by hashing the associated HI.

3. Key Management

Key management aims at guaranteeing the security of cryptographic keys during the period of their application and includes all of the provisions made in a security system design which are related to generation, validation, exchange, storage, safeguard, application, and replacement of cryptographic keys. Appropriate key management is critical to security mechanisms providing confidentiality, entity authentication, data origin authentication, data integrity, and digital signatures. Specifically, a full-fledged key management system should be able to support [Menezes et al. 1996]:

1. Initialization of system users within a domain;

2. Generation, distribution, and installation of keying material;

3. Controlling the use of keying material;

4. Update, revocation, and destruction of keying material; and

5. Storage, backup/recovery, and archival of keying material.

4. Key Revocation

Key revocation is an essential functionality of a security system. By refreshing cryptographic keys, a security system can reduce the dangers of being compromised. Key revocation is also an important step when a security system attempts to confine and recover from the damages caused by attacks. The criteria measuring a key revocation mechanism should include security, efficiency, latency, overheads in terms of communication, etc.

4.1. Background

Cryptographic keys adopted in a security system can be classified into permanent keys and transient keys according to their life periods. As indicated by the name, permanent keys are maintained by holders for relatively long periods which can vary from months to years. Because frequent use of permanent keys can damage their security strength and reduce their valid periods, in many security mechanisms, permanent keys are employed to generate and distribute transient keys which are only valid in relatively short periods (e.g., within a single TCP session). Key revocation issues with transient keys have been taken account of in most authentication mechanisms (e.g., Kerberos, IPSec, SSL, etc.). For instance, in Kerberos, a user can use her password to obtain a session key from a KDC; the session key then can be further used to securely discard and update old sub-session keys. The revocation of transient keys is also considered in the design of HIP. A basic handshaking protocol (i.e., the HIP Base Exchange) has been specified. Using it, two communicating HIP hosts can employ the authenticated Diffie-Hellman algorithm to securely distribute keying material which will be used to generate new cryptographic keys in the following communication. After a handshake, the hosts are able to refresh their transient keys and the corresponding HIP associations, using Update packets.

The revocation issues with permanent keys are also taken into account in lots of key management mechanisms (e.g., PGP, PKI, Peer-to-Peer Key Management for Mobile Ad Hoc Networks [Merwe et al. 2007]). Particularly, in PKI, key revocation issues are addressed in certificate revocation mechanisms.

4.2. Classification of permanent Key Revocation Mechanisms

This draft focuses on the issues with permanent key revocation in HIP. In the remainder of this draft, key revocation indicates permanent key revocation, unless mentioned otherwise.

Mechanisms for key revocation can be classified in various ways, according to:

o Whether additional operations are needed. If a key revocation mechanism does not need any additional operation in the revocation process of a cryptographic key, it is called an implicit key revocation mechanism. The basic idea of an implicit HI revocation mechanism is to associate a key with a valid period and use cryptographic methods to prove the binding between the key and its valid period. Therefore, after the pre-defined period expires, the key is obsolete automatically. For instance, in PKI, a Certificate Authority (CA) can issue a certificate for a user in

order to assert the association between the user and its public key. The certificate is associated with a life period. When the period expires, the user's public key is revoked automatically. If a key revocation mechanism needs to carry out additional operations (e.g., notifications) to revoke a cryptographic key, it is called an explicit key revocation mechanism. In different explicit key revocation mechanisms, such operations can be performed either by a dedicated server or by the owner of the key. Compared with implicit key revocation mechanisms, an explicit key revocation mechanism has the capability to revoke a cryptographic key before its life period expires. For instance, in X.509 [RFC2459] based systems, an issuer can generate a list of certificates, which were revoked for some reasons before their expiring dates, for users to consult.

o Whether a trusted third party is needed. In some revocation mechanisms, the status information of a cryptographic key is provided by a secure third party. A proof of validity is performed during each request from users, and the secure third party provides up-to-date information. Online Certificate Status Protocol (OCSP) for X.509 certificate is such a mechanism. An OCSP client generates an OCSP request that primarily contains the information of one or more queried certificates and send it to a trusted OCSP server. After receiving the OCSP request, the server creates an OCSP response containing the updated status information of the queried certificates. In some other revocation mechanism, validity information is distributed to the requester by a non-secured server. For example, in PGP, a principal can use its revoked key to sign a key revocation certificate and upload it to a key repository server which only provides a repository service and does not make any assertion.

o The list is adopted. According to the information provided, key revocation mechanisms can be classified into black list mechanisms and white list mechanisms. A black list mechanism can provide the information of the keys which are not valid anymore. The Certificate Revocation List (CRL) is an example of this kind of mechanism. In a CRL, revoked certificates are listed in a signed list, so that users can query the information about the revoked keys whenever it is convenient. White list mechanisms, instead, only provide information of valid keys. For example, SSH specify a kind of resource record (RR) called SSHFP [RFC4255]. A SSHFP RR contains the information of the fingerprint of a valid cryptographic key. If a key needs to be revoked, the associated SSHFP RR is removed. If a user cannot find the associated SSHFP RR from DNS, she will believe that the key inquired about is no longer valid.

o The way of distributing revocation information. In a key revocation mechanism applying the push model, when a key is revoked, a server proactively contacts the related users to inform the case. In contrast, in a key revocation mechanism applying the pull model, a client needs to query a server for particular revocation information. OCSP, CRL, and the key revocation mechanisms adopted in PGP and SSH all belong to this category.

There are few discussions about the HI revocation issues with HIP. In the current HIP architecture, hosts are allowed to update their identifiers arbitrarily without notifying others. The lack of HI revocation mechanism can be taken advantage of by attackers to, for instance, escape tracking, bypass ACLs (Access Control Lists), impersonate others using the compromised HIs, etc. In remainder of this document, candidate approaches and related issues are discussed.

5. Implicit HI Revocation in HIP

Implicit key revocation is the most basic key revocation approach. By associating an HI with a life period, the holder of the HI needs to update the HI periodically so as to reduce the risk of HI compromization. In addition, life periods of HIs can help users to verify how long an HI has been used and how long the HI will still be valid. This enables host managers to define more specific security policies.

Note that the HI and the HIT of a host are cryptographically associated. A revocation of an HI will cause the revocation of the corresponding HIT, and vice versa. The life periods of an HI and its HIT are identical; the revocation of a HI implies the revocation of the associated HIT, and vice versa.

The life period of an HI can be specified either by the holder of the HI or by a trusted authority. During HIP BEXs, such life period information can be encapsulated in (to be specified) parameters and transported within HIP packets. If the life period of the HI is specified by its holder, the holder needs to use the associated private key to sign the parameter. If the life period of the HI is specified by a trusted authority, the authority needs to use its private key to sign a life period certificate for the HI. The certificate can be encapsulated within a CERT parameter and transported in HIP packets.

Figure 1 illustrates an example HOST_ID parameter which is extended to transport the associated life period of an HI. This parameter can be applied in the cases where the life period of the HI is specified by its holder. Similar to the life periods of X.509 certificates,

the life period of an HI is specified by a Not Before Time and a Not After Time. In this parameter, the NB Length and NA Length fields indicate the lengths of Not Before Time and Not After Time fields respectively. The Not-Before-Time and the Not-After-Time can be in a format of either UTCTime or GeneralizedTime defined in [RFC2459].

During a HIP base exchange, the parameter containing Initiator' s HI and the associated life period information is transported in the I2 packet, while the parameter containing Responder' s HI and the associated life period information is transported in the R1 packet.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            HI Length          |DI-type|     DI Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            NB Length          |           NA Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               Host Identity                   /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/               |              Domain Identifier                /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                               |          Not Before Time      /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                               |          Not After Time       /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                             |          Padding                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1. An extension of HOST_ID parameter

This approach enables a holder to specify the life period of its HI. It does not rely on any dedicated trusted authority and introduces little performance penalty in verifying the life period. However, this solution is less effective in the environments where communicating HIP hosts lack sufficient trust; it is difficult for a HIP host to identify either the remote host has appropriately defined and managed its HI life period or the HI used by the remote host has not been compromised. In order to reduce memory consumption and foil deny-of-service attacks, HIP hosts normally do not maintain the information of the HIP hosts that they used to communicated with for a long period. In addition, in the current HIP resolution solutions (e.g., HIP RR), no information about the life periods of HIs is provided. If a user of a HIP host assigns a new life period with a reasonable length for the HI before the expiration of the old life period, the update of the life period is unlikely to be detected. Moreover, because HITs are treated by applications as ordinary IP addresses which have no expiration date, in referral scenarios the

receiver of a HIT may not be able to obtain the knowledge of the life period of a HIT from the referrer. In the current HIP resolution solutions (e.g., HIP RR), there is no concern about the life periods of HIs. Therefore, in current HIP architectures, the approach cannot work properly unless there has already been a certain level of trust between two HIP hosts beforehand, that is, a HIP host can believe the HI of its communicating partner is within the declared life period and has sufficient security strength.

The issues mentioned above can be largely addressed by assigning a trusted authority to manage the life periods of HIs and the binding between HIs and HITs. Dedicated trusted auhtorities may introduce complexity into the current HIP architecture, impose additional communications (e.g., registration process, generation of certificate chain, etc.), and cause issues in terms of scalability and trust. However, in many cases they seem to be the only choice. The benefit and the issues brought by dedicated authorities are discussed in section 6 in detail.

The remainder of this sub-section introduces two complementary solutions which are able to mitigate the issues of arbitrarily modifying HI life periods but impose little performance penalty. The first approach is to facilitate the implicit HI revocation functionality with resolution systems systems (i.e., to extend resolution systems to provide trustable life-period information of HIs). For example, the HI life-period information could be maintained by DNS servers and provided to users just like other mapping information. In order to achieve this, space for the life period information needs to be allocated in the resource records sent back to users. In Figure 2, an example extension of the HIP RR with life period information is illustrated. Same as the extended HOST_ID parameter in Figure 1, the NB Length and NA Length fields indicate the lengths of Not Before Time and Not After Time fields respectively. The Not-Before-Time and the Not-After-Time can be in a format of either UTCTime or GeneralizedTime defined in [RFC2459].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| HIT Length    |  PK algorithm |          PK Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            NB Length          |          NA Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              HIT                              /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                               |        Public Key             /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                               |     Rendezvous Server         /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                               |       Not Before Time         /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                               |       Not After Time          /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/       |
+-+-+-+-+
```

Figure 2. An Extension of HIP RR

In this approach, information of the life period of an HI, just like the other information in the RR, can be provided by an authorized user at the registration time. But after the registration, the life period information is only allowed to be updated by the ones who have higher privileges (e.g., server managers). After a user uploads the information of a HIP host in an authoritative DNS server, the user is not allowed to modify the Not Before Time and Not After Time fields of the HI any more. Moreover, after the life period of the HI has expired and is not allowed to be extended, the associated RRs should be removed.

The other approach is to introduce the life period of a HI into the generating process of the associated HIT. For instance, the life period of an HI can be used as a part of the input for generating the associated HIT. Therefore it is computationally difficult even for the holder of the HI to modify the life period without modifying the HIT. For example, after a host advertises its HIP RR, any attempts to modify the life period of the HI can be easily detected, even no life period information is provided by the DNS server. For instance, in the case that a host obtains a HIT from its referrer, it needs to first obtain the knowledge to access the host holding the HIT from resolution servers. Then it can get the associated HI and the life period from the HIT holder, and re-calculate the HIT to verify whether the life period of the HIT is valid. This approach needs little modification on the resolution servers and can be applied independently. A disadvantage of this approach is its inflexibility in the cases where the life periods of HIs need to be extended.

6. Explicit HI Revocation in HIP

As mentioned previously, in many typical scenarios (e.g., the compromise of a key is detected), a cryptographic key need to be revoked before its life period expires. In such cases, explicit key revocation is needed.

When an HI needs to be removed from operational use prior to its originally scheduled expiry, the revocation of the HI needs to be informed to all the hosts which might be affected. If there is no dedicated third party to rely on, the holder of the HI needs to deliver the revocation certificate signed by the associated private key to all the affected partners. The poor scalability of this type of solution is always a subject of debates. First, using this solution, the holder an HI may need to maintain a long list of information about the partners which will be affected by the revocation. Especially when the number of the partners is big, this job can be onerous and error prone. In addition, because HIP does not support multicast, the holder has to generate a notification packet for each of its partners, and send them out during the revocation. When the number of related partners increases, the holder may have to spend a large amount of bandwidth, memory and computing resources in generating and delivering the notification packets. In order to improve the performance of this solution, the holder can send the certificate to a limited set of partners. These partners then relay the certificate to others. However, this solution may introduce additional latency and make the delivery of the certificate un-reliable. Besides the above issues, this solution requires all the involved partners to be online during an HI revocation process, which can be hardly fulfilled on many occasions. Basically, this solution is only suitable in the circumstances where the number of involved hosts is relatively small and stable.

The experiences in PKI demonstrate that pull models can be more scalable in dealing with a large amount of users, and as a result, most of the certification revocation mechanisms (e.g., Certification Revocation Lists (CRLs), delta CRLs [RFC2459], and the On-Line Certificate Status Protocol (OCSP)) proposed in PKI are based on pull models. In these mechanisms, the revocation information is maintained in a third party for users to query whenever it is convenient.

PKI has provided a set of certificate management mechanisms. On many occasions, it is feasible for HIP to take advantage of PKI style solutions to address the issues with HI management.

However, it should be realized that PKI oriented solutions are not silver bullets and cannot be utilized to address all the issues that

HIP has to encounter. After HIP has been globally deployed, it is expected that there will be billions of HIP users which may belong to different organizations and attach to the Internet through different ISPs. Due to the poor scalability of PKI and lack of trust, it is extremely difficult (if possible) to put such a big amount of geographically distributed users under the control of a unique PKI security domain. Therefore, it is reasonable to assume that there will be many different security domains all over the world. When two HIP hosts belong to two different security domains, it may be difficult for a host to verify the assertion made by the security server in the domain of the other one. Although there have been solutions of generating trust relationship across various security domains, all of them impose additional overheads with respect to the construction and verification of credential chain and communication with remote security servers, which negatively influences the performance of HIP. Therefore, the HIP community argues that two HIP-aware hosts should be able to communicate without any additional security facilities. Actually, the only third party server introduced in the base-line HIP architecture is the Rendezvous Server (RVS) [RFC5204]. A RVS only relays messages for the hosts which attempts to communicate with mobile hosts and provides little security functionality. The HIP hosts intending to communicate with each other still need to use the HIP Base Exchange protocol to carry out authentication and exchange keying material for future communications. However, RVSes can be extended to support HI revocation if necessary. When a mobile host changes its HI, it can inform its RVS. Therefore, when the RVS find that a host attempts to access the mobile host with the old HI, the RVS can send the mapping information of the antique HI and the new HI to the host. The RVS needs to use its private key to sign the mapping information in order to ensure the information will not be tampered with. Upon receiving the mapping information, the remote host can use the new HI in the subsequent communications. Additionally, since it is suggested in [RFC5204] that a user get the information of RVSes from DNS, the security of the communication between the remote host and DNS servers needs to be protected. Otherwise, an attacker can easily convince a witness that she is a legal RVS by forwarding a bogus DNS RR consisting of its information to the witness. DNSSEC can be applied to address this issue.

Also, resolution servers can be potentially adopted to construct a global explicit HI revocation mechanism applying a pull model. For instance, when a host intends to revoke its HI, it can send a revocation certificate signed by its private key to an authoritative DNS server. After receiving the certificate, the correspondent RR will be removed, and thus users will not obtain the information about the revoked HI any more. Therefore, DNS servers can perform as a white list HI revocation mechanism, similar to what is specified in

SSH. To avoid the long delay in the spread of revocation information caused by caching RRs on DNS resolvers, the TTL (Time To Life) of RRs can be set to zero. In order to secure the revocation information, DNSSEC should be adopted.

7. Related Discussions

7.1. Influence of HI revocation on Already Generated HIP Associations

In a BEX, HI key pairs of the both communicating partners are used to carry out mutual authentication while the key material for securing subsequent communication are generated by the Diffie-Hellman algorithm. Therefore, if an HI key pair is secure at the time when a HIP association is generated, the later revocation of the HI key pair will not affect the security of the keying material. Assume there is an attacker which has compromised the HI key pair. It is still computationally difficult for the attacker to decrypt the packets transported between the communicating partners. Because the Update packets are under the protection of HMAC, the attacker cannot forge them to interfere with the communication. Note that the attacker can try to forge Notify packets. However, according to [RFC 5201] Notify packets are only informative, which will not affect the state of the communicating partners. Therefore, if no explicit key revocation occurs, the expiry of an HI does not have to affect the security strength of HIP associations generated using the HI when it is still valid. They still can be used until they reach their expiring time. However, if an HI is found to be compromised, the security of the keying material of the already generated HIP associations cannot be guaranteed. In practice, the compromise of a cryptographic key can be perceived only after the attacks employing the key are detected. It is difficult for one to identify the exact time from which the key is no longer secure. Hence, under this circumstance, the pre-generated HIP associations can only be used to deliver revocation certificates, as it is difficult for the communicating partners to know whether the HI is still secure when the HIP associations were generated.

7.2. HI Refreshment

In key management mechanisms, key refreshment is concerned with the issues of using new cryptographic keys to take place of "old" ones. Therefore, it closely related with key revocation. A refreshment procedure of a key can occur either before or after the revocation of the key (Note that in the first case the key is still valid). In this section, the issues with HI refreshment in HIP are discussed.

Ideally, an operational HI should be refreshed before its crypt-

period is expired. In thsi case, the holder can use the old HI to establish secure channels, and use Update packets to transport the refreshment information to related partners (in a push model) or to trusted third parties (in a pull model). In the Update packets, the new HI and other related information are encapsulated. Therefore, before the old HI expires, both HIs are valid, and the HIP associations generated with the old HI can still be applied.

In practice, the third parties deployed for HI revocation can also be used to support HI refreshment. For instance, when using a pull model, a host can transport the HI revoking and the refreshing information to a third party. Therefore, when a user inquires of the third party about the status information of an HI, the user can get the status of the HI inquired about as well as the associated refreshment information.

If an HI needs to be revoked due to accident disclosure or compromise, the update of the HI can be a little more complex. Although the invalid key can be used to send a "suicide" information to others (e.g., resolution systems, RVSes, or any entities which may be affected by the revocation), it cannot be used to securely transport the refreshment information any more.

If a host has multiple HIs, it can select a HI still valid to securely transport the refreshment information. The refreshment information should consist of both the new HI and the compromised HI. This solution requires that the partner communicating with the host can ensure that the HI used to generate secure channel and the compromised HI are possessed by the same HIP host. Such knowledge can be obtained from resolution systems or provided by the host. It is recommended that there is a HI used only for HI refreshment.

In the cases where all the HIs of a host become invalid (e.g., the host is found to compromised), the host only can distribute the refreshment information using an out-of-band way.

A host can also implement a pull model by directly transporting the update information to resolution servers. If the information is forwarded to a DNS server, users can query the latest HI using FQDN of the host. In a resolution system providing ID to locator mapping services (e.g., DHT), users can only try to query the resolution systems using old HITs. In this case, besides the IP addresses inquired, the resolution system should also provide the latest HIs and other useful information. Note that it is assumed that no two HITs of different hosts are identical, even if they are adopted in different time period. In practice, because the length of HITs is long, the possibility that two hosts select a same HI can be very low. In order to further reduce the possibility, a user can also

provide the life period of the inquired HIT to the resolution server.

8. Conclusions

Key revocation is critical for HIP to be secure, practical and manageable. Particularly, HIP hosts are expected to keep working securely for a relatively long period, proper key revocation mechanisms for HIs must be provided. This document focuses on pros and cons of different key revocations and analyzes their security and practicality in different practical scenarios. Although key management has been an active research area for a long period and lots of successful key-management systems (e.g., PKI) are widely adopted in practice, many issues (e.g., scalability, lack of trust) still exist. There is no solution being found to meet the timeliness and performance requirements of all applications and environments that HIP is expected to support [McDaniel et al. 2001]. Therefore, it is predicted that various HI revocation approaches will be adopted after HIP has been globally adopted.

9. IANA Considerations

This document makes no request of IANA.

10. Security Considerations

The important of HI revocation can be various according to the usage of HITs. When HITs are used for authentication/access control, the HI revocation is critical to prevent attackers from using compromised HIs to access certain resources illegally. In the scenarios where HIP is purely used as an ID/Locator separation solution to support mobility or multi-homing and the authentication issues are addressed by other security mechanisms, the HI revocation is less important.

In the existing HIP architectures, the HI of a HIP host acts as both the identifier and the public key of the HIP host at the same time. The revocation of the host's public key will result in the change of the identifier of the host. Without the assistance of other measures, the host will be regarded as a different one by others. The instability issue introduced by the HI revocation must be considered in designing identity management and resolution systems for HIP hosts. For instance, during the revocation of a HI, all the TCP sessions identified with the associated HIT have to be broken. There are two solutions can be considered in addressing this problem.The first one is to check the life period of a HI before using it to construct a TCP session and guarantee that the HI can be

stable during the communication. The second one is to introduce a stable identifier to represent a HIP host for up layer protocols. The new identifier should not have to be changed during the update of a HI.

11. Acknowledgements

Many Thanks to Thomas.R.Henderson for his kindly revision and precious comments.

12. References

12.1. Normative References

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

[RFC2459] Housley, R., Ford, W., Polk, T., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC5204] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 5204, April 2008.

[RFC5205] Nikander, P. and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", RFC 5205, April 2008.

12.2. Informative References

[McDaniel et al. 2001]
McDaniel, P. and A. Rubin, "A Response to "can we eliminate certificate revocation list?"", 2001.

[Menezes et al. 1996]
MENEZES, A., VAN OORSCHOT, P., and S. AND VANSTONE, "Handbook in Applied Cryptography", 1996.

[Merwe et al. 2007]
Merwe, J., Dawoud, D., and S. McDONALD, "A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks", 2007.

[Recommendations]

Barker, E., Barker, W., Burr, W., Polk, W., and M. Smid, "Recommendation for Key Management-Part1-General(Revised)", March 2007.

Authors' Addresses

Dacheng Zhang
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di Information Industry Base, Hai-Dian District
Beijing, 100085
P. R. China

Phone:
Fax:
Email: zhangdacheng@huawei.com
URI:

Dmitriy Kuptsov
Helsinki Institute for Information Technology
PO. Box 9800, TKK FI-02015
Finland

Phone:
Fax:
Email: dmitriy.kuptsov@hiit.fi
URI:

Sean Shen
CNNIC
4, South 4th Street, Zhongguancun
Beijing, 100190
P.R. China

Phone:
Fax:
Email: shenshuo@cnnic.cn
URI: