IBE Extension for HIP

Amir K. C.[1], Harri Forsgren[2], Kaj Grahn[3], Timo Karvi[2] and Göran Pulkkis[3]

[1]Aalto University, Espoo, Finland
[2]University of Helsinki, Helsinki, Finland
[3]Arcada University of Applied Sciences, Helsinki, Finland

Abstract. This article explores the possibilities to replace RSA public key identities and X.509 certificates with any unique identities and identity-based encryption (IBE) in the Base Exchange of the Host Identity Protocol (HIP). We have analysed the technical and trust-related details when applying IBE in HIP. These details include, for example, how to insert the IBE parameters into HIP packets and how to guarantee their correctness. We have extended OpenHIP v0.7 software with capabilities for X.509 certified RSA-based Host Identities, for trusted IBE-based Host Identities, and for IBE signatures in HIP messages. We have also measured HIP message times in the Base Exchange. These measurements show that the basic IBE solution is rather slow compared to RSA solution with certificates. However, if applications are such that it is necessary to check revocation lists often, the IBE solution is feasible.

1 Introduction

Recently, there have been many suggestions how to modify or extend the TCP/IP protocol stack in order to increase security and trust between communicating partners ([1], [2], [3], [4], RFC 3972). One of these developments is the Host Identity Protocol (HIP). The basic idea of HIP is to separate addresses or domain names from network host identities. In HIP, an identity is defined by a RSA public key pair. In the connection setup phase public keys are used to identify the participants and to verify digital signatures in data packets. However, trustworthy verification of an RSA signature requires certification of the signer's public key. In addition for chained certification, signature verification is needed for every certificate in the chain and the most recent revocation list must be checked. Thus verification of a signature created by a certified identity can be quite time consuming.

In this paper, we explore the possibilities to replace certified RSA identities by IBE identities (Identity Based Encryption). IBE has a key escrow feature, because a Private Key Generator (PKG) knows also private client keys. Thus IBE may not be appropriate in applications, where communicating participants belong to different organizations and countries. But inside one large organization IBE could be a better option than certified RSA identities. We present the techniques and modifications needed, if IBE is applied in HIP. We have also made concrete time measurements in order to find out, if IBE is a better alternative than RSA.

K.C. A., Forsgren H., Grahn K., Karvi T. and Pulkkis G..
IBE Extension for HIP.
DOI: 10.5220/0004601700440052
In *Proceedings of the 10th International Workshop on Security in Information Systems* (WOSIS-2013), pages 44-52
ISBN: 978-989-8565-64-8

2 Identity-based Cryptography

In identity-based cryptography (IBE) a public key is derived from identifying information like email address, phone number or domain name. The corresponding private key is created by a private key generator (PKG) that must be trusted by all participants. The technique of IBE is based on elliptic curves and pairings on these curves.

2.1 Elliptic Curves

In elliptic curve cryptography, a curve must generate a sufficiently large additive group. A scalar multiplication of a point P with an integer a is denoted as $a.P$. In our measurements, we have used a super singular curve with with embedding degree 2 and the Barreto-Naehrig curve. These curves are defined in the Annex A of [5] and have also been implemented in the software package PBC version 0.5.11 as curve A and F, respectively. [6]

The curve A is defined over a finite field $\mathbb{F}_p$, where p is a 512 bit prime. We also need for the curve A a cyclic subgroup, whose order is of size 160 bits. The curve F is defined over a finite field $\mathbb{F}_q$, where q is 160 bits. Only such elliptic curves are suitable for cryptography for which the problem in Definition 1 hard.

Definition 1 (Computational Diffie-Hellman Problem CDH). *Given a group G and $P, a.P, b.P \in G$, compute $ab.P \in G$.*

2.2 Pairings

Let G_1 and G_2 be additively denoted abelian groups, practically groups defined by elliptic curves, and G_3 a multiplicatively denoted cyclic group. Let $\varphi : G_2 \to G_1$ be a distortion mapping (see [5]). Cryptographic pairing $e : G_1 \times G_2 \to G_3$ is a bilinear function that is non-degenerative i.e. $e(\varphi(P), P) \neq 1$ for all generators $P \in G_2$ and the Bilinear Diffie-Hellman Problem (Definition 2) is hard for it.

Definition 2 (Bilinear Diffie-Hellman Problem BDH). *Given P, $a.P$, $b.P$, $c.P \in G_2$, compute $e(\varphi(P), P)^{abc} \in G_3$.*

3 Host Identity Protocol

With HIP (RFC 5201) an IPsec security association (SA) can be established. Moreover, HIP supports mobility with a mechanism to update host's IP address without breaking an established IPsec SA.

3.1 HIP Base Exchange

In HIP Base Exchange (see Fig. 1) all HIP packets should include Host Identity Tags (HIT) of both peers. HIT is a 128-bit SHA-1 hash of the Host Identity (HI). The initiator of the connection starts with an I1 packet. Once the responder receives the I1 packet, it doesn't need to store any state information related to the session that is about to be established. In order to continue, the initiator must solve a cryptographic puzzle sent in R1. Packets R1 and I2 contain data for a Diffie-Hellman session key. Packets I2 and R2 have part of their contents encrypted with the session key.

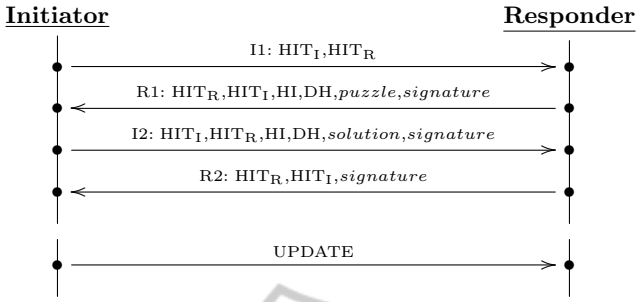

Fig. 1. HIP Base Exchange.

3.2 HIP Security and Trust

There are a number of security and trust problems associated with mobility and multihoming. The responder's HIT must be available to the initiator before HIP Base Exchange, since the opportunistic mode, which allows I1 without the responder's HIT, is vulnerable to man-in-the-middle attacks. Pre-calculation of R1 and the puzzle mechanism protect the responder against Denial of Service (DoS) attacks in HIP Base Exchange. Diffie-Hellman key agreement vulnerability to man-in-the-middle attacks is eliminated in HIP Base Exchange by signing packets R1, I2, and R2. Communication confidentiality is established by currently recommended ESP encryption of payload data. HIP includes various protections against malicious replays. Pre-signed R1 messages protect against replays of I1 packets, and the puzzle mechanism and optional use of opaque data protect against false I2 packets. R1 packets have a monotonically increasing generation counter, whose value is kept across system reboots and invocations of the HIP Base Exchange. Packet authenticity is ensured by HMACs and digital signatures. HMAC is verified with a key derived from a Diffie-Hellman session key. Especially low-powered hosts can rely on trusted middle boxes verifying digital signatures. In HI certification for trust in HIs, the HIP CERT parameter can be used in all HIP packet as a container for certificates. (RFC 5201, 5202, 5206, and 5207)

3.3 HIP Implementations

Current HIP implementations are OpenHIP [12], which is used in this paper, InfraHIP [13], Hip4Inter [14], PyHIP [1], and CuteHIP [15, 16]. OpenHIP is an open source and free software implementation of HIP developed by Boeing Company. The latest release of OpenHIP application is OpenHIP 0.9 released on March 2012. OpenHIP supports Linux, Windows and OS X environments. OpenHIP supports DNS extensions of HIP and is capable of fetching HIP data from Domain Name Server (DNS). Additionally, the Linux version of OpenHIP has built-in Rendezvous Server.

4 Implementation of IBE Extension for HIP

In this paper peers are allowed to have their own PKGs with different parameters. The

private user key computed by the PKG from the public user key must be sent to the user via a secure channel. The IETF standard RFC 5408 describes how the private and public information is sent from a PKG to a user. PKG certification guarantees that the right PKG is used but it does not guarantee that the secret key is safe. Key escrow in IBE can be avoided for example by using threshold techniques in distributed generation of private user keys with multiple PKGs ([11]). Some protection against disclosure of a private user key can be obtained by changing public user keys and corresponding private user keys reasonably often.

With IBE, the HI can be just host's domain name. This is an advantage over RSA and over RSA based digital signatures, since anyone is able to construct a HIT for an I1 packet by just knowing the host's domain name. Alternatively, a regularly changing IBE host identity could be appended with a date code encoded in a standardized way.

A normal R1 packet defined in RFC 5201 contains the HI, Diffie-Hellman value, a proposal for encryption algorithm, HMAC algorithm, digital signature algorithm, a puzzle, a HMAC and a digital signature. The IBE extension uses an IBE digital signature algorithm instead of RSA.

4.1 Choice of an IBE Signature Scheme

Many signature schemes using Identity Based Cryptography (IBE) have hitherto been proposed ([7]). The IEEE draft standard [5] has adopted a signature scheme proposed in [9] because of its efficiency compared to other schemes like in [8]. Let A, B be peers and let A sign a message m. Let PKG be A's private key generator. PKG chooses and publishes the following parameters:

- Abelian groups G_1, G_2, and G_3 of prime order p. In practice, the groups G_1 and G_2 are additive groups on elliptic curves and G_3 is a multiplicatively denoted group.
- A distortion map $\varphi : G_2 \to G_1$.
- Generators $Q \in G_2$ and $P = \varphi(Q) \in G_1$.
- A pairing $e : G_1 \times G_2 \to G_3$ and the value $g = e(P, Q) \in G_3$.
- A public key $Q_{pub} = sQ \in G_2$, where $s \in \mathbb{Z}_p^*$ is a randomly chosen master key.
- Hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_p^*$ and $H_2 : \{0,1\}^* \times G_3 \to \mathbb{Z}_p^*$.

The signer A generates his private key $K_A = (H_1(ID_A) + s)^{-1}Q \in G_2$. He also chooses a random integer r. The signature for a message m is (h, S), where $h \in \mathbb{Z}_p^*$ and $S \in G_1$. These are calculated as follows:

$$u = g^r,$$
$$h = H_2(m, u),$$
$$S = (r + h)K_A.$$

The receiver B verifies the signature by computing

$$u' = e(S, H_1(ID_A) \cdot P + Q_{pub})e(P, Q)^{-h},$$

and accepts the signature if $h = H_2(m, u')$.

The signer A has calculated the pairing operation beforehand, so the only heavier operation is the power calculation $u = g^r$. The verifier must, on the other hand, compute both pairing and power.

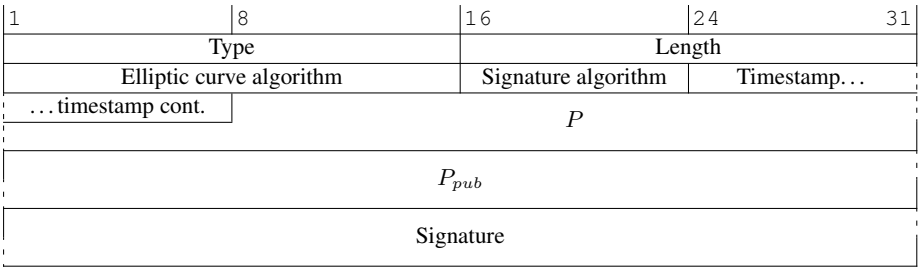

Fig. 2. HIP parameter structure for PKG public parameters.

4.2 Trust in Public PKG Parameters

In order to check a signature in an R1 packet, the initiator must know the public parameters of the receiver's PKG. These public parameters are included in the R1 packet in a new parameter part, whose structure is shown in Fig. 2. The part contains information on the elliptic curve algorithm in use and public points P and P_{pub}. The field for the elliptic curve algorithm has an identifier for an elliptic curve formula, its coefficients, field $\mathbb{F}_p$ and sizes of subgroups generated by P and $\varphi(P)$. The parameters used in this article are defined in Section 4.1. There is a timestamp that tells, when the PKG should publish a new signed public parameter block. The packet is signed with a traditional PKI signature algorithm like RSA-SHA1. PKG public parameters signature can be verified once the responder has sent all required certificates.

A peer must gain trust in a PKG's validity. An active attacker could create a fake PKG for anyone. A trust is derived from the use of certificates to ensure that the PKG has authority over the HI. Certificate issued to the PKG must include information on its intended purpose to be used to authenticate hosts in a given domain. Once verified, the public PKG parameter signature doesn't have to be checked again until the parameters have expired. A revoked PKG certificate or availability of only expired PKG parameters could imply that PKG's private key s might be compromised.

5 Performance Measurements

HIP Base Exchange times have been measured for RSA HIs with and without X.509 certificates and for IBE HIs with X.509 certified public PKG parameters. For the measurements, OpenHIP v0.7 has been deployed with necessary additions and modifications needed for X.509 certificates, service-offer parameters, and IBE HIs with IBE signatures specified in the IEEE draft standard in [5] using PBC Library[6]. See [17] for details. Standard X.509 certificates are created with OpenSSL. A self-signed root certificate is first created and thereafter some levels of certificates. A first level certificate is signed by the root certificate, a second level certificate is signed by a first level certificate, etc.

Our measurement setup was a home PC connected to a host in Arcada University of Applied Sciences via an operator gateway, the FUNET WAN, and a gateway in

Arcada. In HIP Base Exchange with X.509 certified HIs, the Initiator sends two certificates and the Responder sends 2, 4, 5 and 7 certificates. The Responder sends a SERVICE_OFFER in R1 but no certificates. This way unnecessary certificates won't be delivered, unless the initiator specifically asks for them. The actual certificates are sent in R2 and in UPDATE packets after R2. The Initiators certificates come in I2, which also contains a SERVICE_OFFER parameter. The certificates sent form a certificate chain. Certificate revocation is not checked. In HIP Base Exchange with IBE HIs, the Initiator and the Responder used different PKGs with X.509 certified public parameters.

Table 1 shows for X.509 certified HIs the time average, standard deviation and maximum/minimum values of ten HIP Base Exchanges, when Responder sends zero, two, four, five, and seven certificates. Four certificates are sent in R2 and in one UPDATE packet, five certificates need two UPDATE packets, and seven certificates need three. For the unmodified OpenHIP v0.7 similar measurements, without certificates, gave the results 62 ms for Initiator and 62 ms for Responder. It turned out that the number of certificates does not affect much on the time spent in Base Exchange. This was not expected, because extra certificates cause extra packets and extra rounds in Base Exchange. Maybe the time differences would be longer, if the network distances were longer. Now the distances were inside one metropolitan area.

Table 1. HIP Base Exchange times in milliseconds.

Number of certificates	Initiator				Responder			
	Avg.	Std. dev.	Max.	Min.	Avg.	Std. dev.	Max.	Min.
0	68	5	81	63	72	5	78	60
2	64	6	79	59	68	6	75	56
4	68	6	79	63	72	5	74	59
5	71	6	83	66	75	6	80	63
7	70	3	80	70	73	3	77	67

Table 2 shows processing times for individual HIP packets. The times for (Initiator,I1) and (Initiator,I2) include the transmission delay for sending I1/I2, processing time of I1/I2 by Responder, and the transmission delay for receiving R1/R2. Similarly, the times for (Responder,R1 with certs), (Responder,R1 without certs) and (Responder,R2) include the transmission delays.

Table 2. Processing time of individual HIP packets in milliseconds.

	I1	R1 with certs	R1 without certs	I2	R2	UPDATE
Initiator	4	33	41	27	3	3
Responder	< 1	58	64	3	4	3

For HIP Base Exchange with IBE signatures packet construction times, packet processing times and network delay time were measured as is shown in Fig. 3. For example, R1 construction time is $t_3 - t_2$, R1 processing time is $t_5 - t_4$, and total network delay is $(t_1 - t_0) + (t_4 - t_3) + (t_7 - t_6) + (t_{10} - t_9)$.

Table 3 shows for IBE HIs the time average, standard deviation and maximum/ minimum values of ten HIP Base Exchanges for IBE signatures based on Elliptic Curve A in [6] and Table 5 shows the same for Elliptic Curve F in [6]. The measured time

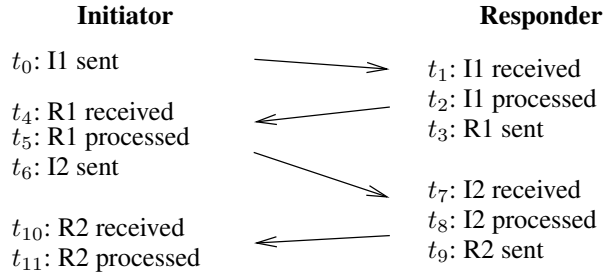

Fig. 3. Measured time values t_0, t_1, t_2,..., t_{11} in HIP Base Exchange with IBE signatures.

values in Tables 3, 4, 5 and 6 include times for exchange of certificates for the public PKG parameters of both the Initiator and the Responder, but certificate revocation check times are not included. As Table 1 shows, leaving certificates out would not reduce times essentially.

Table 3. The Base Exchange time using A curve in milliseconds.

	Initiator	Responder	Packet construction	Network delay
Avg.	242	233	97	7
Std. dev.	77	77	75	1
Max.	425	416	279	9
Min.	161	153	23	6

Table 4. Processing time of individual HIP packets using A curve in milliseconds.

	R1 construction	R1 processing	I2 construction	I2 processing	R2 construction	R2 processing
Avg.	5	86	91	46	6	7
Std. dev.	< 1	7	75	11	2	2
Max.	5	97	273	78	11	13
Min.	5	70	17	40	5	6

The most secure and probably also fastest revocation check of a X.509 certificate issued by a known trusted CA is an online check, for example using Online Certificate Status Protocol — OCSP (RFC2560). An online revocation check is more secure than a local offline check, since there is no vulnerable time window between a revocation list update and the local availability of an updated revocation list. Online revocation check is probably also faster than a local offline check, since a CA is expected to maintain revocation lists in a quite efficient computer, and therefore the local offline revocation check time may often be longer than the sum of the online check time and the network traversal time. Measured OCSP based certificate revocation check times are on average 790 ms for server certificates issued by Verisign and 660 ms for a server certificate issued by Thawte.

Verification of a X.509 signature should include a revocation check. The measured HIP Base Exchange times in Tables 1, 3 and 5 should therefore be prolonged with online revocation check times both for X.509 certified HIs and IBE HIs trusted by X.509

Table 5. The Base Exchange time using F curve in milliseconds.

	Initiator	Responder	Packet construction	Network delay
Avg.	611	530	88	7
Std. dev.	52	52	51	1
Max.	722	643	196	9
Min.	546	470	39	5

Table 6. Processing time of individual HIP packets using F curve in milliseconds.

	R1 construction	R1 processing	I2 construction	I2 processing	R2 construction	R2 processing
Avg.	15	214	72	226	214	78
Std. dev.	< 1	10	52	6	10	4
Max.	17	231	181	236	231	84
Min.	15	198	24	218	198	74

certified public PKG parameters. However certificate verification with revocation check is much more frequently required for X.509 certified HIs than for IBE HIs. Certificate verification with revocation check is required for an X.509 certified HI for each new connection. For IBE HIs X.509 certificate verification with revocation check is required in HIP Base Exchange only when a HIP communication party uses a PKG, which is hitherto unknown to the HIP communication partner.

6 Conclusions

Current HIP specifications support only public RSA and DSA keys for Host Identities (HI). Trust in a responder HI can be derived in HIP Base Exchange from DNSSEC when HIP DNS Extension (RFC5205) is deployed. Trust in HIs can also be derived from X.509 certification of public RSA or DSA keys used as HIs. [10]

Public IBE keys would be natural HI option, since network host names and identity tokens representing network host users could be used as HIs to exclude the need to certify HIs for trust. Deployment of IBE HIs however requires extension of current HIP specifications as well as a solution to the private key escrow problem in IBE and trust in the PKG, which generates and delivers private IBE keys. The solution to the private key escrow problem based on threshold cryptography and distributed private key generation in multiple PKGs [11] however leads to considerable implementation complexity.

A trust solution for a PKG is X.509 certification of the public PKG key [10]. Since current specifications for X.509 certification (RFC3279, RFC4055, RFC4491) do not support public IBE keys this trust solution requires a PKG to include a supported key pair in public key cryptography, for example a RSA key pair, for certification of the public PKG key and other public PKG parameters. Performance measurements for an OpenHIP v0.7 implementation with the PBC Library [6] of this trust solution indicate, that HIP Base Exchange is at least about 3 times slower than for a trust solution based on X.509 certification of HIs, when certificate revocation is unchecked. Trusted IBE HIs

may still be computationally more efficient than X.509 certified HIs, since our measured average times of online certificate revocation check are longer than our measured average HIP Base Exchange times for trusted IBE HIs, and since certificate verification with revocation check is required far more frequently for X.509 certified HIs than for trusted IBE HIs.

References

1. Gurtov, A.: Host Identity Protocol (HIP): Towards the Secure Mobile Internet. Wiley (2008)
2. Andersen, D.G., et al.: Accountable internet protocol (aip). In: Proceedings of SIGCOMM 2008. (2008) 339–350
3. Camarillo, G., Melen, J.: HIP (Host Identity Protocol) Immediate Carriage and Conveyance of Upperlayer Protocol Signalling (HICCUPS) (2010) Internet Draft.
4. Lagutin, D.: Securing the Internet with Digital Signatures. PhD thesis, Aalto University, Department of Computer Science and Engineering, Espoo, Finland (2010)
5. P1363 Working Group: IEEE P1363.3™/D1 Draft Standard for Identity-based Public-key Cryptography Using Pairings. IEEE. (2008) Retrieved October 19th, 2008 from http://grouper.ieee.org/groups/1363/IBC/material/P1363.3-D1-200805.pdf.
6. PBC library Retrieved January 10, 2013, from http://crypto.stanford.edu/pbc/.
7. Dutta, R., Barua, R., Sarkar, P.: Pairing-based cryptographic protocols: A survey. Cryptology ePrint Archive, Report 2004/064 (2004) Retrieved November 15th, 2012 from http://eprint.iacr.org/2004/131.
8. Hess, F.: Efficient identity based signature schemes based on pairings. In: SAC 2002. Volume 2595 of LNCS., Springer-Verlag (2002) 310–324
9. Barreto, P., Libert, B., McCullagh, N., Quisquater, J.J.: Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Asiacrypt. Volume 3788 of LNCS., Springer-Verlag (2005) 515–532
10. Forsgren, H., Grahn, K., Karvi, T., Pulkkis, G.: Security and trust of public key cryptography options for HIP. In: 10th IEEE International Conference on Computer and Information Technology (CIT 2010), IEEE Conference Publications (2010) 1079–1084
11. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. SIAM Journal of Computing 32(3) (2003) 586–615
12. OpenHIP overview (March 2012) Retrieved November 25, 2012 from http://www.openhip.org/wiki/index.php?title=Overview.
13. Helsinki Institute for Information Technology: HIP for Linux (2004) Retrieved November 25, 2012 from http://infrahip.hiit.fi/index.php?index=about.
14. Ericsson Ab, NomadicLab: HIP for inter.net project (2008) Retrieved November 25, 2012 from http://hip4inter.net/.
15. CuteHIP project Retrieved November 24, 2012 from http://code.google.com/p/cutehip/.
16. Kuptsov, D.: Implementing CuteHIP: Feasibility analysis of Java-based network-layer security protocols. Technical report, Aalto University (2011) Retrieved November 24, 2012 from http://www.hiit.fi/u/kuptsov/resources/cutehip.pdf.
17. Modified OpenHIP 0.7 Retrieved February 3, 2013, from http://www.cs.helsinki.fi/u/hforsgre/openhip.