An Identity-Based Key Agreement Protocol for the Network Layer

Christian Schridde, Matthew Smith, and Bernd Freisleben

Department of Mathematics and Computer Science, University of Marburg
Hans-Meerwein-Str. 3, D-35032 Marburg, Germany
{schriddc,matthew,freisleb}@informatik.uni-marburg.de

Abstract. A new identity-based key agreement protocol designed to operate on the network layer is presented. Endpoint addresses, namely IP and MAC addresses, are used as public keys to authenticate the communication devices involved in a key agreement, which allows us to piggyback much of the security overhead for key management to the existing network infrastructure. The proposed approach offers solutions to some of the open problems of identity-based key agreement schemes when applied to the network layer, namely multi-domain key generation, key distribution, multi-domain public parameter distribution, inter-domain key agreement and network address translation traversal.

1 Introduction

Current network security protocols like IPsec use either pre-shared keys or a Public Key Infrastructure (PKI) to secure the communication channel. The pre-shared keys approach is suitable for small networks but does not scale well. The PKI approach scales better but has a high management overhead [27],[2]. To avoid the complexity of authenticated public key distribution, Shamir [25] in 1984 proposed the concept of identity-based cryptography (IBC) which allows an arbitrary string to be used as a public key. Since then, several identity-based encryption (IBE) schemes [16] and identity-based key agreement protocols [14] have been suggested, but it was not until 2001 when the first practical IBE systems were introduced by Boneh and Franklin using Weil pairings [6] and Cocks using quadratic residues [10].

IBC has been applied to several application layer protocols, with the main focus lying on e-mail protection. Some attempts have been made to apply IBC to lower layers like the network layer to offer lightweight alternatives to PKI based security solutions, but a number of problems have hindered the adoption of IBC at this level. Unlike the application layer where identifiers are usually unique (like e-mail or SIP addresses) and do not change owners, IP addresses can both change owners on a regular basis and are not necessarily unique. For both scarcity and security reasons, many devices have private IP addresses and access the Internet using the Network Address Translation (NAT) protocol. This creates problems for IBC, since outside the private network the device behind the NAT router has a different identifier, namely that of the NAT router. Furthermore, the private identifier of the "NATed" device is most likely used by other resources in other private networks and thus is not unique. Another practical issue when deploying

R. Ostrovsky, R. De Prisco, and I. Visconti (Eds.): SCN 2008, LNCS 5229, pp. 409–422, 2008.

an IBC system on the network layer is that many different organizations are responsible for different parts of the Internet and thus it is unlikely that a single trusted authority can be found to operate the identity private key generator (ID-PKG). Several solutions have been proposed which allow multiple ID-PKGs to interoperate [15,17,7,5], but these systems require either cooperation between the ID-PKGs or a hierarchical approach with a trusted party at the top. Both approaches are difficult to use in the Internet due to organizational difficulties and conflicting business interests. As demonstrated by approaches based on a Certificate Authority (CA), there will always be competing organizations offering the same service for the same protocol (e.g. signing RSA public keys) without wanting to cooperate on the corporate level. Thus, to successfully deploy IBC on the network layer, the IBC system must be able to cope with NAT, address reuse (and consequently dynamic identity key deployment), and it must allow competing organizations to operate their ID-PKG independently of other ID-PKGs while still enabling cross-domain execution of the IBC protocols for their customers.

In this paper, a new identity-based key agreement protocol is introduced which focuses on the issues to be solved when implementing IBC on the network layer. The proposed approach is realized using IP addresses on the network layer and optionally MAC addresses on the data link layer for bootstrapping purposes. It utilizes the mathematics also used in the traditional Diffie-Hellman [12] key agreement and Rivest-Shamir-Adleman (RSA) [23] public key cryptography approaches, and in the key distribution system proposed by Okamoto [19]. Solutions to the problems of multi-domain key generation, key distribution, multi-domain public parameter distribution, cross-domain key agreement and NAT are presented.

The paper is organized as follows. Section 2 discusses related work. In Section 3, the new identity-based key agreement scheme is presented. Section 4 discusses implementation issues. Section 5 concludes the paper and outlines areas for future research.

2 Related Work

Since Shamir's pioneering IBC proposal in 1984, various IBE systems have been proposed. In 2001, Boneh and Franklin [6] introduced an operational IBE system based on bilinear pairings on elliptic curves. By using efficient algorithms to compute the pairing function, the system can be used in real world applications. Its main application focus is e-mail, dealing with key distribution and expiration in this domain. Several optimizations and extensions of Boneh and Franklin's IBE approach have been suggested (e.g. [17], [8], [9], [26]), all based on the Weil pairings [6] originally used by Boneh and Franklin. These extensions include hierarchical IBE systems [15] and public parameter distribution systems [27]. However, the main focus of all proposals is on application level security, and e-mail is the main application.

Apart from the full IBE systems, several identity based key agreement schemes have been proposed, such as [14], [19], [17], [24], [9], [26] and [8]. For example, the key distribution system proposed by Okamoto [19] extracts identity information and combines it into a session initiation key in a similar manner as in our scheme, but does not address the problem of key agreement between different domains. Other of these approaches offer solutions to combine a number of ID-PKGs, but not in an independent manner.

Appenzeller and Lynn [2] have introduced a network layer security protocol using IBC. The protocol does not deal with the setup phase, i.e. public parameter distribution, identity key distribution, and is not compatible with NAT routers. In [27], a working solution for the problem of public parameter distribution on an Internet scale using the Domain Name System (DNS) has been presented. However, the paper does not deal with the critical issues of identity key distribution or NAT traversal.

In all of the above proposals, an arbitrary string can be used as an identifier and thus also as a public key. However, since the main application focus of most of these proposals is e-mail, e-mail addresses are the only identifiers for which problems and solutions are discussed in depth. Problems occurring in other areas, such as IP, Voice-over-IP, NAT traversal, multi-provider networks and dynamic IP redistribution, are not discussed. Furthermore, although hierarchical Identity Private Key Generator (ID-PKG) systems have been introduced [15,17,7,5], they usually require either a root authority trusted by all ID-PKGs, or that the different ID-PKGs cooperate during their setup. This is problematic both in the Internet and in telecommunication networks, since no single trusted authority exists, and service providers are typically competitors.

Two other approaches have been suggested in the context of secure IPv6 network protocols that are relevant to this paper. The Host Identity Protocol (HIP) [18] removes the need for binding IP addresses to public keys using certificates by creating a completely new form of addresses, namely HIP addresses which are constructed by hashing the public key of a device. This creates two requirements which a HIP system must meet: (a) the public keys must be created before the address allocation can be performed and (b) a new protocol layer must be implemented between the transport and network layer which maps HIP identifiers to the routable IPv6 addresses and provides authentication. To address the last critical issue, Cryptographically Generated Addresses (CGA) [4] have been propsed to encode a public key into the 64-bit identifier of the IPv6 address, thus avoiding the need to change the protocol stack. However, CGA still requires the public key to be created before the IPv6 address and restricts the choice of addresses which can be used. Obviously, getting ISPs to issue particular IPv6 addresses based on user keys is a difficult task.

3 A New Identity-Based Key Agreement Scheme

3.1 Requirements

The requirements that should be met by the proposed key agreement protocol for the network layer are as follows:

- Complexity reduction: Contemporary networks are already complex environments with thousands of network enabled devices coming and going, roaming in foreign networks and changing providers on a continuous basis. A security infrastructure should not double the administrative effort by creating one or more mirrors of the administrative infrastructure for public key certification or storage. At some point, keys or public parameters must be distributed if authentication is required, but these key management issues should be kept as local as possible while offering global interaction.

- Multi-provider operation: The Internet is driven by multi-organizational entities which will not adopt a single trusted third party or a single trust hierarchy. Therefore, a cryptographic system must allow multiple security providers to inter-operate with minimal requirements.
- NAT traversal: NAT is a common occurrence in the Internet, and even with the adoption of IPv6 it will not disappear completely, since many organizations use NAT to protect networked resources as well as to conserve public IP addresses.
- Dynamic/transient endpoint addresses: Due to the relatively small number of IPv4 addresses, these endpoint addresses are shared or reused after a certain amount of time. A cryptographic system must be able to deal with changing owners of endpoint addresses.

To combine the ease of use of a Diffie-Hellman key agreement protocol with the security of an authenticated RSA key exchange, we present an identity-based key agreement protocol in which the endpoint addresses of the communication devices are used to authenticate the devices involved in a key agreement. The use of IBC allows us to both reduce and spread the administrative burden of key management by seamlessly integrating the cryptographic solution into the existing network infrastructure. Our scheme allows multiple organizations to operate ID-PKGs autonomously, while allowing their customers to operate across organizational borders. This greatly simplifies the otherwise thorny issue of private key distribution present in global IBE or PKI solutions. Furthermore, the choice of coupling the cryptographic system to the network layer allows us to piggyback much of the security overhead to the existing network infrastructure. The proposed system is capable of coping with NAT traversal, and using key expiration coupled with dynamic key deployment allows for dynamic endpoint allocation.

In the following, our identity-based key agreement system called Secure Session Framework (*SSF*) is presented. First, the four basic steps of the system are described for the single ID-PKG scenario. Then, the scheme is extended to incorporate multiple autonomous ID-PKGs which use different public parameters without requiring entities from one ID-PKG's domain to contact the ID-PKGs of other domains.

3.2 Algorithmic Overview

The proposed identity-based key agreement protocol *SSF* consists of four main algorithms: **Setup**, **Extract**, **Build SIK**, and **Compute**. The first two are executed by the private key generator and the last two are executed by a participating device. Their description follows below.

3.3 Key Agreement

The **Setup** algorithm is executed by the ID-PKG. This part of the key agreement protocol is only performed once and creates both the master secrets P and Q as well as the public parameters.

Setup - The **Setup** algorithm is executed by the ID-PKG.
Input: $k \in \mathbb{N}$
Step 1: Choose an arbitrary integer $R > 1$ from $\mathbb{Z}^+$.
Step 2: Generate two primes, P and Q, of bit length k with the following properties:
1. The prime factorization of $(P-1)$ contains a large prime P'
2. The prime factorization of $(Q-1)$ contains a large prime Q'
3. $(\varphi(PQ), R) = 1$, where $\varphi(\cdot)$ is the Totient Function.
4. The computation of discr. log. must be infeasible in $\mathbb{Z}_P$ and $\mathbb{Z}_Q$.
Step 3: Compute the product $N = PQ$
Step 4: Choose a generator G of a subgroup $\mathbb{G}$ of $\mathbb{Z}_N$ whose order contains at least one of the primes P' or Q'.
Step 5: Choose a cryptographic collision-resistant hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_{\mathbb{N}}$.
Output: $PSP = (N, G, R, H(\cdot))$, $SP = \{P, Q\}$

Definition 1 (Public, Shared Parameters). *The public, shared parameters (PSP) of a domain* ***D*** *of the key agreement scheme* SSF *is the quadruple* $PSP = (N, G, R, H(\cdot))$

The **Extract** algorithm creates the identity key (i.e. the private key) for a given identity. This algorithm is executed by the ID-PKG. If all IDs are known and the range is not too big (e.g. a Class B or C subnet of the Internet), it is possible to execute this step for all IDs offline, and the master secrets can then be destroyed, if required.

Extract - The **Extract** algorithm is executed by the ID-PKG.
Input: PSP, SP, ID
Let ID be a given identity. The algorithm computes $d_{ID} \equiv H(ID)^{1/R} \; mod \; N$. The integer d_{ID} is called the *identity key* and is given to the entity E_{ID}.
Note: Due to the requirement $(\varphi(N), R) = 1$, the R-th residues form a permutation in $\mathbb{Z}_{\mathbb{N}}$.
Ouput: d_{ID}

The **Build SIK** algorithm is executed by the devices taking part in the key agreement.

Build SIK - The **Build SIK** algorithm is executed by the entity E_{ID}
Input: PSP, d_{ID}, k_1, k_2
Step 1: Choose a random integer r_{ID} in the interval $[2^{k_1}, 2^{k_2}]$.
Step 2: Compute $SIK_{ID} \equiv G^{r_{ID}} \cdot d_{ID} \; mod \; N$.
SIK_{ID} is the SIK (session initiation key) for the identity string ID which belongs to entity E_{ID}.
Output: SIK_{ID}

The random integer r_{ID} is generated with a secure number generator to make r_{ID} unpredictable. The private identity key is used in combination with this randomly chosen integer and the generator in such a way that it is not possible to extract the identity key from the SIK. This is due to the fact that the multiplications are performed in the ring $\mathbb{Z}_{\mathbb{N}}$ and the result set of a division in the ring $\mathbb{Z}_{\mathbb{N}}$ is so large that the extraction of the

identity key is infeasible. The SIK is then sent over an unsecured channel to the other party and vice versa. The SIK must be greater than zero to prevent a trivial replacement attack where an attacker replaces the SIKs with zero which in turn would make the session key zero as well. Any other replacement attacks lead to invalid session keys.

The final step of the key agreement process is the computation of the session key using the **Compute** algorithm which is executed by the devices taking part in the key agreement. By applying the inverse of the hash value of the opposite's identity, the involved identity key is canceled out. Only if both endpoint addresses match their identity keys, a valid session key is created.

Compute - The **Compute** algorithm is computed when two entities are performing a key agreement.
Input for E_{ID_A}: PSP, $SIK_{ID_B} > 0, ID_B, r_{ID_A}$
Input for E_{ID_B}: PSP, $SIK_{ID_A} > 0, ID_A, r_{ID_B}$
When E_{ID_A} receives the session initiation key from E_{ID_B}, it calculates
$(SIK_B^R H(ID_B)^{-1})^{r_{ID_A}} \equiv ((G^{r_{ID_B}} d_{ID_B})^R H(ID_B)^{-1})^{r_{ID_A}} \equiv G^{Rr_{ID_A} r_{ID_B}} \equiv S \; mod \; N$
When E_{ID_B} receives the session initiation key from E_{ID_A},it calculates
$(SIK_A^R H(ID_A)^{-1})^{r_{ID_B}} \equiv ((G^{r_{ID_A}} d_{ID_A})^R H(ID_A)^{-1})^{r_{ID_B}} \equiv G^{Rr_{ID_A} r_{ID_B}} \equiv S \; mod \; N$
Output: S

3.4 Key Agreement between Different Domains

The ID-PKG determines the public, shared parameters, and all entities which receive their identity key for their IDs from this generator can establish a key agreement among each other. In practice, it is unlikely that all devices will receive their identity key from the same security domain, since this would imply the existence of a third party trusted by all with a secure communication link to all devices. The Internet is divided into many Autonomous Systems (AS) each with its own IP range and responsible for the management of its users. Thus, it is desirable that each AS can operate its own ID-PKG.

In this section, we show how cross-domain key agreement can be achieved such that only the public parameters must be distributed (which will be discussed in section 4). Each device only needs a single identity key, and the ID-PKGs do not need to agree on common parameters or participate in any form of hierarchy. In the following, we assume without loss of generality, that there are two domains D_1 and D_2. Their public parameters are $(N_1, G_1, R_1, H_1(\cdot))$ and $(N_2, G_2, R_2, H_2(\cdot))$, respectively. Every parameter can be chosen independently. The case that $(R_2, \varphi(N_1)) > 1$ or $(R_1, \varphi(N_2)) > 1$ is not critical, since no R-th roots must be computed regarding the other domain's modulus. The two moduli N_1 and N_2 were chosen according to the requirements stated in the **Setup** algorithm, i.e. the computation of discrete logarithms is infeasible in $\mathbb{Z}_{N_1}$ and $\mathbb{Z}_{N_2}$, respectively. Consequently, an algorithm such as the Pohlig-Hellman algorithm [20] cannot be applied and Pollard's $P-1$ factoring algorithm [21] will not be a threat. Thus, a random non-trivial integer has a large order in $\mathbb{Z}_{N_1 N_2}$ with an overwhelming probability, and the computation of discrete logarithms is infeasible in $\mathbb{Z}_{N_1 N_2}$. In the following, an entity E_{ID_A} from D_1 wants to communicate with E_{ID_B} from D_2.

Cross-Domain Extension (from the view of entity E_{ID})
Input: PSP_1, PSP_2, d_{ID}
Step 1: Calculate the common, shared, public parameters: $PSP_{1,2} = (N_1 \cdot N_2, G_1 \cdot G_2, R_1 \cdot R_2, H_2(\cdot))$.
Step 2: Use the Chinese-Remainder Theorem to calculate the integer $\widetilde{d_{ID}}$: $\widetilde{d_{ID}} \equiv d_{ID} \; mod \; N_1$ and $\widetilde{d_{ID}} \equiv 1 \; mod \; N_2$
Step 3: Use the Chinese-Remainder Theorem to calculate the integer $\widetilde{H_1}(ID)$: $\widetilde{H_1}(ID) \equiv H_1(ID)^{R_2} \; mod \; N_1$ and $\widetilde{H_1}(ID) \equiv 1 \; mod \; N_2$
Output: $SIK_{ID}^{(1,2)}$

In step 1 of the cross-domain extension algorithm, the common shared public parameters are the element-wise product of both sets of domain parameters. In step 2, entity E_{ID_A} extends its identity key using the Chinese-Remainder Theorem. In step 3, entity E_{ID_A} extends its hash identifier also using the Chinese-Remainder Theorem.

The procedure for entity E_{ID_B} is analog, only the indices change from A to B. Key agreement is then performed using the following extension of the original algorithm.

Cross-Domain: Compute algorithm
Input for E_{ID_A}: $PSP_{(1,2)}$, $SIK_{ID_B}^{(1,2)} > 0$, ID_B, r_{ID_A}
Input for E_{ID_B}: $PSP_{(1,2)}$, $SIK_{ID_A}^{(1,2)} > 0$, ID_A, r_{ID_B}
When E_{ID_A} receives the session initiation key from E_{ID_B}, it calculates
$\left(((G_1G_2)^{r_{ID_B}}\widetilde{d}_{ID_B})^{R_1R_2}\widetilde{H_2}(ID_B)^{-1}\right)^{r_{ID_A}} \equiv (G_1G_2)^{R_1R_2r_{ID_A}r_{ID_B}} \equiv S \; mod \; (N_1N_2)$
When E_{ID_B} receives the session initiation key from E_{ID_A},it calculates
$\left(((G_1G_2)^{r_{ID_A}}\widetilde{d}_{ID_A})^{R_1R_2}\widetilde{H_1}(ID_A)^{-1}\right)^{r_{ID_B}} \equiv (G_1G_2)^{R_1R_2r_{ID_A}r_{ID_B}} \equiv S \; mod \; (N_1N_2)$
Output: S

A security analysis, correctness proofs and further details on the algorithms can be found in [3].

4 Implementation Issues

In the following, several issues for deploying the proposed system in practice are discussed. It will be shown how the public parameters and the identity keys are distributed in multi-provider scenarios and how dynamic IP addresses are handled. Furthermore, a detailed description of how our system deals with the NAT problem will be given. One of the important issues of any multi-organizational cryptographic system is the distribution of the public parameters and keys. It should be noted that a main requirement is to try to minimize the number of global distribution steps in favor of local distribution steps, since this distributes the workload and reduces the risk of a global compromise. In a scenario with N providers, each with M customers where $M >> N$, we have $N \cdot M$ customers in total. This means that $N \cdot M$ private/identity keys need to be distributed. In a PKI, in the worst case in which everybody wants to communicate with everybody else, $(N \cdot M - 1) \cdot (N \cdot M)$ public keys need to be exchanged and managed. In our system,

only the public parameters of the N providers need to be exchanged. This reduces the number of transfers from $N \cdot M$ local and $(N \cdot M - 1) \cdot (N \cdot M)$ global transfers to $N \cdot M$ local transfers and only N global transfers, and since $M >> N$, this is a large saving. Even using traditional key distribution mechanisms, our system offers a significant saving compared to a PKI in key escrow mode. In the following, further optimizations of the distribution process which are possible due to the network centric approach of our solution will be suggested.

4.1 Distribution of Shared, Public Parameters

Like most other IBC approaches, our system also uses shared public parameters. In a single domain scenario, the distribution of the public parameters is straightforward. However, if each AS runs its own ID-PKG, the number of public parameters and the binding between public parameters and identity keys becomes more complex. As stated above, this distribution problem is still much smaller than the distribution problem for traditional public keys where each entity has its own public key which needs to be distributed. Of course, traditional PKI technology can be used to distribute the public parameters, but a more suitable solution is to integrate the public parameters into the DNS lookup messages. In this way, the fact that a DNS lookup is made anyway to resolve a host IP is utilized, and the public parameter transfer can be piggybacked to the DNS reply. The technical details of the integration of IBC public parameter information into DNS records were evaluated by Smetters and Durfee [27]. Their positive evaluation lead us to adopt the public parameter distribution technique for our system. For more information on the details of how to incorporate this kind of information into the DNS system, the reader is referred to [27], [1] or [13]. To secure the transport, either DNSsec can be used or the public parameters can be signed and transferred with standard DNS, or a key agreement can be executed between the requesting party and the DNS server if the public parameters of the DNS server are known. Since the DNS server is usually in the same AS as the requesting customer, this is not a problematic issue, because the public parameters are the same as the customer's public parameters. As stated above, this part of the system has been tried and validated by several research groups.

4.2 Distribution of the Identity Keys

The most critical element in all IBEs or PKIs in key escrow mode is the distribution of identity keys (private keys) and the prevention of identity misbinding. In traditional PKI and IBE systems, this is usually done manually and out-of-band and thus creates a lot of work. While it can be argued that due to the fact that on the AS level most customers receive an out-of-band message when they receive their endpoint address, adding a fingerprint to the identity key would not put much extra burden on the system. However, a more elegant solution for the long term is to integrate the key distribution into the IP distribution system. For most networks, this means integration into the DHCP server. This, however, is not trivial since DHCP on its own is an unsecured protocol not suitable for transferring private information. The two main threats are packet sniffing and MAC spoofing. If the identity key is sent in the clear via the DHCP protocol in an unswitched network, an attacker can sniff the identity key, leading to key compromise. With MAC

spoofing, an attacker pretends to be the legitimate owner of a foreign MAC address, and the DHCP server sends the identity key to the attacker. Both forms of attacks make the plain use of DHCP for key distribution infeasible.

In the following, we present several solutions geared towards different scenarios of how the distribution of identity keys can be integrated into DCHP securely. In a fixed corporate network environment using a switched infrastructure, the easiest solution is to use the MAC lockdown function of modern switches. Using MAC lockdown, each port gets a MAC address and will only serve that MAC address. Thus, if an attacker wishes to spoof a MAC address to gain the key, physical access to the correct port must be acquired, significantly increasing the risk and effort of the attack. This scenario works in a corporate network where each MAC address is registered and assigned to a port anyway. In a student dormitory, for example, it is less feasible since managing the ever changing MAC addresses of the private devices used by students would be very time consuming and error prone. Here, an IEEE 802.1X + Radius [11,22] solution is more practical. The authorization is usually done in the form of a user-name password check. The IP address and the corresponding identity key can either be fixed (as set by the Radius and DHCP server) or dynamic and transient (more on transient IP addresses in section 4.3 and 4.4). Either way, only the legitimate user receives the identity key, and it is not possible to spoof the MAC address to receive a copy in the same key lifetime. If packet sniffing is an issue, the DHCP request needs to be extended to include a protected session key with which the identity can be protected from sniffing attacks. The client creates a session key which is encrypted using the public parameter N (N can be used in the same way as an RSA public key) of the key generator of the DCHP server and broadcasts the DHCP request. The session key can only be decrypted by the DHCP server which then uses the session key to encrypt the identity key of the client, using e.g. the Advanced Encryption Standard AES, which is then broadcasted. Thus, the identity key can only be decrypted by the client.

Apart from these two practical solutions based on an extension of existing security mechanisms which can be used in the short term, we also present a more speculative long term solution which does not rely on other security mechanisms. In this case, we bootstrap the network layer key agreement scheme on the data link layer by using MAC addresses as public keys. As with IP addresses, we cannot assume that there will be a single authority to generate the MAC identity keys, but since our system does not require cooperation between the ID-PKGs, this can be handled. Each organization with the authority to distribute MAC addresses runs its own ID-PKG and writes the identity key onto the networking card at the same time as the MAC address. Since the MAC addresses are globally unique and should not change over the lifetime of the networking card, a fixed identity key is not a problem. On the contrary, a hardware based protection of the key creates an added layer of security. Organizations with the right to distribute MAC addresses have their own Organizationally Unique Identifier (OUI) which is encoded in the first three octets of all MAC addresses distributed by this organization. Using this OUI, the public parameters needed for the MAC address can be found. This entails a very small and lightweight public parameter lookup mechanism matching OUIs to public parameters. This is the only step where some form of cooperation is needed on the organizational level, since all OUIs must be publicly available.

However, since the number of OUIs is small and does not change frequently, it is easy to solve this part of the distribution. The huge benefit of this structure is that the identity key distribution can now be automated in-band in a secure fashion without relying on extensive existing security mechanisms. Using this approach, it is possible for the requesting entity to add a proof of legitimate MAC address possession using the identity key of the MAC address when requesting its IP address. This not only prevents the problem of MAC spoofing, but also allows the DCHP server to send the identity key for the IP address to the requesting entity protected with the MAC based identity encryption. Since this mechanism is only used for requesting the identity key, which is done in an Intranet, the proposed solution does not open a backdoor to the Network Interface Card producers to decrypt the Internet traffic.

4.3 Key Expiration

Another practical issue of network layer encryption is the fact that especially in IPv4 networks, IP addresses are reused. In a PKI or CA based IPsec solution, this creates several problems, since the central PKI must be updated or the CA must be contacted to resign public keys as the users swap IP addresses. Certificate Revocation Lists can be used to accomplish this, but the response time until a change is propagated is quite long and creates a fair amount of effort. In particular, public key caching mechanisms can lead to problems. In our identity-based solution, natural key expiration techniques can be used to cope with dynamic IP addresses. Boneh et al. [6] showed how keys can be given a lifetime, which allows natural expiration of the identity key. This is done by the concatenation of the ID, in our case the IP address, with a date. The same technique can be used in our solution. In the scenario where ISPs have a pool of IP addresses which are allocated to customers on demand and reused at will, this technique can be used such that no two customers ever receive the same identity key. Since IP address reuse is time-delayed in any case[1], this time frame can be used as the key lifetime to ensure that each successive owner lies in a new lifetime slot. With the techniques introduced in this paper, a frequent automatic in-band key distribution can be safely executed and thus key renewal is far less of a problem. Additionally, key expiration also reduces the risk of identity key theft, since the attack window is restricted to a small time interval.

4.4 Transient Key Generation

If a large network requires the uses of transient keys, the key generator can be required to generate a large number of identity keys. To ease the computational load of the key generator, we implemented an extension to the generation protocol which makes the generation of identity keys less computationally expensive, at the cost of making the session key calculation more expensive. This distributes the load from a single point (the key generator) to a large number of resources (the clients). The extension makes uses of the fact that exponentiation using repeated squaring is faster when the binary representation of the exponent is a sparse-one integer. A sparse-one integer D of order

[1] Before an IP address is allocated to a new user, a certain amount of time must pass to prevent attackers from impersonating the previous entity.

δ is an integer whose binary representation has exactly δ 1's. The extended algorithm is presented below.

Extract - The **Extract** algorithm is executed by the ID-PKG.
Input: PSP, SP, *ID*
Let *ID* be a given identity. The algorithm computes $d_{ID} \equiv H(ID)^D \ mod \ N$.
Choose D as a sparse-one integer. The PSP are increased by the value E from $D \cdot R \equiv E \ mod \ \varphi(N)$.
Output: d_{ID}

There are exactly $\binom{k}{\delta}$ sparse-one integers of order δ with bit-length k. Thus, δ should be selected to be sufficiently large to make exhaustive search infeasible.

4.5 NAT Traversal

The final practical issue is the NAT problem. While this mainly is a problem in IPv4 networks, there are also scenarios in IPv6 networks in which NAT is an issue. The main problem when dealing with network layer encryption when NAT is involved is that the NAT server substitutes its public IP address for the private IP address of the entity being NATed. As such, the original identity key for the private IP address is no longer valid, since it does not match the public IP address of the NAT router, hence any key agreement would fail. This problem is also faced by IPsec which has problems with NATed resources. When working in a NAT environment, a certain level of trust must exist between the NAT router and the NATed device. The NAT router substitutes its public IP address for the private IP address of the NATed device. The NATed device must trust the NAT router to substitute the right address, and the NAT router must be willing to forward the packets on behalf of the NATed device. However, when using encryption, the NATed device does not trust the NAT router with the plain text version of its communication. Communication between the NATed device and the outside world should still be private. Considering that the NAT router shares its public IP address with the NATed devices, our solution also lets the NAT router share the identity key of its public IP address with the NATed devices (we will show later that this does not compromise the security of either the NAT router or the NATed devices). The identity key of its Intranet IP address is, however, kept private. Also, a private identity key is given to each NATed device, corresponding to its Intranet IP address. When a NATed device A in the Intranet establishes a connection to an external device B, it creates a SIK packet using its private value G^{r_A} in combination with the identity key of the NAT router's public IP address. This is, in essence, an extension of the normal NAT procedure to include an authenticated key exchange, and the trust relationship between the NAT router and the NATed device is not changed. The sharing of an identity key belonging to an IP address is not usual and should be avoided under normal circumstances, since anyone in possession of the identity key can pose as the legitimate owner of the corresponding IP address and thus can spoof the address or act as a man-in-the-middle attacker. However, in the NAT scenario this is exactly the desired outcome, since the NATed devices pretend to be the NAT router to the outside world, since as far as the outside world is

concerned, the packets originate from the NAT router. It is important to note that although the identity key of the NAT routers' public IP address is used by the NATed device, the NAT router is not able to subvert the communication. To successfully attack the communication as a man-in-the-middle, the NAT router would also need to be in the possession of the private identity key of *B*, which is not the case. It is also not critical if more than one device is behind the same NAT router, since communication between the NATed devices and the NAT router is protected by the private identity key of the NAT router's Intranet IP address and the identity key of the NATed device, which is different for each device. Thus, the NATed devices are not able to subvert the communication of other devices in the Intranet nor are they able to spoof the internal identity of the NAT router or other NATed devices. Should the Intranet devices be connected to the NAT router with a pre-configured switch, the Intranet identity keys are not necessary, since the private value G^{r_A} of the key agreement is sufficient to protect the key exchange if there is a direct connection to the NAT router.

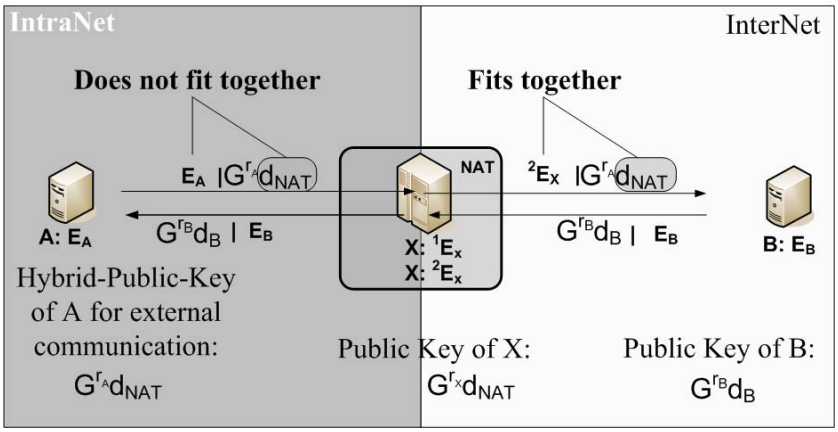

Fig. 1. Proposed solution for the NAT problem with endpoint keys

Figure 1 shows the solution for the NAT problem. The internal user *A* sends a SIK using its own private value G^{r_A} in combination with the private key of the NAT router's IP address. When the NAT router substitutes the IP address with its own, it creates a valid packet, since the value d_{NAT} now belongs to the correct source address of the packet.

5 Conclusions

In this paper, a new identity-based key agreement protocol has been presented to secure the communication channel between two devices using their endpoint addresses as public keys for authentication. The proposed approach has been demonstrated using IP addresses on the network layer and MAC addresses on the data link layer to bootstrap the system, which has allowed us to piggyback much of the security overhead for

key management to the existing network infrastructure. Unlike other identity-based encryption solutions, the presented approach is based on the well tested mathematics also used in the traditional Diffie-Hellman key agreement and Rivest-Shamir-Adleman public key cryptography approaches instead of elliptic curves or quadratic residues. It has been shown how our identity-based key agreement protocol can be used as a generic low level security mechanism and how it can deal with the implementation issues of multi-domain key generation, key distribution, multi-domain public parameter distribution, key expiration, inter-domain key agreement and network address translation traversal.

There are several areas of future work. For example, a more detailed description of the integration of the proposed identity-based approach into existing network management protocols and tools, in particular the integration into the DHCP protocol, should be provided. Furthermore, the large-scale practical deployment of the proposed approach in IP, Voice-over-IP, or mobile telephone communication scenarios is an interesting area for future work.

References

1. Adida, B., Chau, D., Hohenberger, S., Rivest, R.L.: Lightweight Email Signatures (Extended Abstract). In: 5th International Conference on Security and Cryptography for Networks, pp. 288–302 (2006)
2. Appenzeller, G., Lynn, B.: Minimal-Overhead IP Security Using Identity Based Encryption, Technical Report, Voltage Inc. (2002)
3. Schridde, C., Smith, M., Freisleben, B.: An Identity-Based Key Agreement and Signature Protocol with Independent Private Key Generators. Technical Report, Dept.of Mathgematics and Computer Science, University of Marburg, Germany (2008)
4. Aura, T.: Cryptographically Generated Addresses, RFC 3972 (2005)
5. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
6. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. SIAM Journal of Computation 32(3), 586–615 (2003)
7. Boyen, X., Waters, B.: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
8. Chen, L., Cheng, Z., Smart, N.P.: Identity-based Key Agreement Protocols from Pairings. International Journal of Information Security 6(4), 213–241 (2007)
9. Chen, L., Kudla, C.: Identity Based Authenticated Key Agreement Protocols from Pairings. In: 16th IEEE Computer Security Foundations Workshop (CSFW 2003), p. 219 (2003)
10. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Proceedings of the 8th IMA International Conference on Cryptography and Coding, pp. 360–363. Springer, Heidelberg (2001)
11. Congdon, P., Aboba, B., Smith, A., Zorn, G., Roese, J.: IEEE 802.1X Remote Authentication Dial. In: User Service (RADIUS) Usage Guidelines, RFC 3580 (September 2003)
12. Diffie, W., Hellman, M.E.: New Directions In Cryptography. IEEE Transactions On Information Theory (6), 644–654 (1976)
13. Fenton, J., Allman, E., Libbey, M., Thomas, M., Delany, M., Callas, J.: DomainKeys Identified Mail (DKIM) Signatures, RFC 4870 (2007)

14. Günther, C.G.: An Identity-Based Key-Exchange Protocol. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 29–37. Springer, Heidelberg (1990)
15. Horwitz, J., Lynn, B.: Toward Hierarchical Identity-Based Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
16. Maurer, U.M., Yacobi, Y.: A Non-Interactive Public-Key Distribution System. Designs, Codes and Cryptography 9(3), 305–316 (1996)
17. McCullagh, N., Barreto, P.: A New Two-Party Identity-Based Authenticated Key Agreement. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 262–274. Springer, Heidelberg (2005)
18. Moskowitz, R., Nikander, P., Jokela, P., Henderson, T.: Host Identity Protocol, RFC 4423 (October 2003)
19. Okamoto, E.: Key Distribution Systems Based on Identification Information. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 194–202. Springer, Heidelberg (1988)
20. Pohlig, S.C., Hellman, M.E.: An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance. IEEE Trans.on Info. Theory IT-24, 106–110 (1984)
21. Pollard, J.: Theorems of Factorization and Primality Testing. Mathematical Proceedings of the Cambridge Philosophical Society 76, 521–528 (1974)
22. Rigney, C., Rubens, A., Simpson, W., Willens, S.: Remote Authentication Dial In User Service (RADIUS), RFC 2138 (April 1997)
23. Rivest, R.L., Shamir, A., Adleman, L.: A Method For Obtaining Digital Signatures And Public-Key Cryptosystems. Communications Of ACM 1(2), 120–126 (1978)
24. Sakai, R., Kasahara, M.: ID based Cryptosystems with Pairing on Elliptic Curve. In: Symposium on Cryptography and Information Security (2003)
25. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
26. Smart, N.P.: Identity-based Authenticated Key Agreement Protocol based on Weil Pairing. Electronics Letters 38(13), 630–632 (2002)
27. Smetters, D.K., Durfee, G.: Domain-based Administration of Identity-Based Cryptosystems for Secure E-Mail and IPSEC. In: SSYM 2003: Proceedings of the 12th Conference on USENIX Security Symposium, Berkeley, CA, USA, p. 15. USENIX Association (2003)