Dynamic, Non-Interactive Key Management for the Bundle Protocol

William L. Van Besien
Johns Hopkins University Applied Physics Lab
11100 Johns Hopkins Road
Laurel, Maryland, USA
William.Van.Besien@jhuapl.edu

ABSTRACT

Secure, low-overhead key establishment is crucial to maintaining the high level of trust and security that are required many types of Delay Tolerant Networks. Existing schemes for key negotiation and exchange that are currently in use on the Internet often cannot scale to meet the environmental and technical constraints of many Delay Tolerant Networks. The few works presenting solutions to DTN key establishment have largely focused on targeted networking environments. This paper proposes a dynamic, and non-interactive scheme to facilitate secure communication in infrastructureless networks, supporting various levels of trust. Specifically, the solution presented in this paper provides a key management solution to opportunistic overlay networks using the Bundle Protocol.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Cryptographic Controls—*Key Management*

General Terms

Security

Keywords

Key Management, Security, Delay Tolerant Networking

1. INTRODUCTION

Securing a Delay Tolerant Network, e.g., a deep-space network representing multiple space agencies, poses an interesting set of constraints that cannot be easily solved by methods currently in use on the Internet. End hosts may vary dramatically in their computational capabilities, reachability, and levels of trust. Cryptographic suites have to be carefully chosen to balance strong security guarantees with efficient implementation. While reference implementations of the Bundle Protocol security extensions have become available, key management in particular remains an open issue.

The application of a Public Key Infrastructure (PKI) or other negotiation-based key exchange to many challenged networks is impractical because of the environmental constraints. The reachability of key authorities within a fixed time frame cannot be assumed, and round trip negotiations could involve an arbitrarily long delay. Many self-organizing key establishment schemes employed in Mesh Ad-Hoc Networks (MANETs) that rely on probabilistic topology inferences and frequent exchanges [7, 8, 9] may introduce more uncertainty than many network deployments may be willing to tolerate. This necessitates the requirement for a non-interactive key establishment scheme to minimize nondeterminism, overcome environmental challenges, address the relationships between cooperating groups, and support dynamic network growth.

This paper suggests a key distribution scheme for challenged networks using the Bundle Protocol, which can produce cryptographic keys for each of the three security mechanisms detailed in the Bundle Security Protocol specifications. It draws from recent results in bilinear mappings over elliptic curves to enable most pairs of members within a DTN to non-interactively - and based only on the identifier of the remote entity - establish a shared secret from which cryptographic keys can be derived.

CHANTS'10, September 24, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0139-8/10/09.

2. OVERVIEW

2.1 Foundations

The scheme presented in this paper is predicated on recent cryptographic constructions using bilinear maps over elliptic curves. More detailed, formal descriptions of the schemes can be found at the Pairing Based Crypto Lounge [2]. For the purposes of this paper, the following premises must be understood: (1) $e : G_1 \times G_1 \rightarrow G_2$, e maps two elements in one elliptic curve group to one element in a second cyclic group, (2) e is symmetric: $e(X, Y) = e(Y, X)$, (3) The Discrete Logarithm problem remains intractable in G_2 (i.e., solving for s in $e(A, B)^s$ is hard), and (4) e is an efficiently computable operation.

A suitable pairing, e, and elliptic curve groups G_1, G_2, can enable non-interactive key establishment by adhering to the following method (assume g is a group administrator):

1. g selects a random $s_g \in \mathbb{Z}_q$, where q is a prime and the order of the elliptic curve group G_1.

2. Prior to deployment, g pre-loads each constituent host, X, with $S_X = s_g X$. X is an element in the elliptic curve G_1, and is calculated by hashing the *identity* of X (e.g., a Bundle Protocol identifier) into the elliptic curve group.

3. For two hosts, A and B, to non-interactively establish a shared secret, k_{AB}, then A calculates $k_{AB} = e(S_A, B)$.

Note this is equivalent to $e(sA, B) = e(A, B)^s$. Likewise, B calculates $k_{BA} = e(A, S_B)$. Since e is symmetric, $k_{AB} = k_{BA} = e(A, B)^s$, and the two hosts have a distinct pairwise symmetric secret.

The reader is referred to [1, 3, 4, 5, 6], and especially [2] for a more encompassing overview of symmetric key establishment using bilinear maps over elliptic curves.

3. KEY ESTABLISHMENT FOR BSP

This section details the proposed protocols for establishing keys for each of the three security mechanisms described by the BSP specifications. In order to simplify integration of our scheme into existing Bundle Protocol-enabled networks, we assume that this key management scheme will be used to derive keys for HMAC-SHA1 authentication, RSA digital signatures, and AES encryption.

The scheme assumes a many-to-many mapping of hosts and groups[1]. A host can be a member of multiple groups, and contains a "key ring" of secret-shares, one for each group. Only hosts with a group in common can establish a shared secret. We expect there to be two basic types of groups: proximity-based groups and ownership-based groups. Proximity based groups may be run by some international consortium, such that any member can establish shared secrets with any other host in its proximity. This is useful to securely exchange routing or other network management information. Ownership-based groups, in contrast, are used to define keys for securing application data, and these groups are likely to be defined by hosts with a common mission or purpose.

3.1 BAB Key Protocol

The Bundle Authentication Block is used to authenticate Bundles between two neighboring hosts using a symmetric key. Let l be the size, in bits, of the nonce.

1. Select group-id, g, that both A and B have in common.
2. Select $r \in \{0, 1\}^l$ to be used as nonces.
3. Calculate the shared secret: $k_{AB} = e(S_A, B)$
4. Derive symmetric key $K_{AB} = KDF(k_{AB}, r)$
5. Calculate MAC for message, $Auth_{K_{AB}}(M)$
6. A sends the tuple $(g, r, M, Auth(M))$
7. B receives the tuple, calculates $s_{AB} = e(A, S_B)$, derives the key, and checks $Auth(M)$

The parameters containing key information (the group-id and nonce) can be stored in the "Key Information" and "Salt" fields, respectively, that are allocated in the BAB packet headers but are currently unused. For the purposes of key derivation and message authentication, the NIST-certified PBKDF (RFC 2898) and HMAC-SHA1 algorithms are used, respectively.

[1]Though in practice no more than two groups - three for gateway nodes - should be assigned to a single host.

3.2 PIB Key Protocol

The Payload Integrity Block is used to append RSA digital signatures for end-to-end message integrity. We assume that in many challenged networks, interaction with a key- or certificate-authority is impractical, and all identity credentials must be available "up-front". Thus, the protocol in section 3.1 can be extended to authenticate RSA public keys in the following way:

1. g generates its own key pair (Pu_g, Pr_g), and a pair (Pu_i, Pr_i) for each host $i \in g$
2. Each host, i, is assigned the tuple $(Pu_i, Pr_i, Pu_g, Enc_{Pr_g}(Pu_i))$, the final item is the host's public key encrypted with the private key of g.

Since each host has a copy of its public key encrypted with the group authority's private key, its authenticity can be verified by any other host in the group. Thus, for some entity A wishing to send B an RSA-signed PIB, A sends the tuple:

$$(Enc_{Pr_g}(Pu_A), M, Sign_{Pr_A}(M)).$$

This method maintains the property that the integrity signature is checked at every opportunity, since every host in g is able to decrypt $Enc_{Pr_g}(Pu_A)$. Neither can A nor an attacker create an arbitrary public key, as all keys must be signed by the group authority, g. The receiving host, B, calculates ...

$$Pu_A^? := Dec_{Pu_g}(Enc_{Pr_g}(Pu_A))$$

If $checkSig(Pu_A^?, M, Sign_{Pr_A}(M))$, **then** accept.
Else reject.

Recall that $Enc_{Pr_g}(Pu_A)$ is pre-loaded onto host A. For added security, the following field may be added that is checked *only* on the receiving end-host.

$$(Enc_{Pr_g}(Pu_A), Auth_{K_{AB}}(Enc_{Pr_g}(Pu_A)), M, Sign_{Pr_A}(M)).$$

The second field in this tuple is the authentication of the signature of A's public key, and could be useful in the case that the group's RSA key pair has been compromised.

3.3 PCB Key Protocol

The Payload Confidentiality Block (PCB) encrypts the Bundle's contents using AES encryption with a symmetric key. The PCB uses a master Key Encryption Key (KEK) to encrypt a temporary Bundle Encryption Key (BEK). The PCB can be represented as the following tuple $(Enc_{KEK}(BEK), M, Enc_{BEK}(M))$. BSP assumes the KEK is a pre-distributed symmetric secret. Again, the BAB protocol from section 3.1 can be extended to produce a symmetric key for the PCB.

1. Select group-id, g, that both A and B have in common.
2. Select $r \in \{0, 1\}^l$ to be used as nonce.
3. Set $k_{AB} = e(S_A, B)$
4. Derive $KEK = KDF(k_{AB}, r)$
5. A sends the tuple $(g, r, Enc_{KEK}(BEK), M, Enc_{BEK}(M))$ (*note: the BEK is generated within the PCB functionality*)

4. PROTOCOL DYNAMICS

The security of the proposed scheme ultimately rests upon the intractability of the Discrete Logarithm problem over G_1 and G_2. For large exponents and well-studied curves this provides strong assurances, but nevertheless this assurance degrades over time[2]. In order to maintain a fresh pool of secrets, we assume there exists some suitable parameter for each group T_{g_i} (the "latency diameter") within the DTN. This represents the maximum time needed for the authority of each group to contact the most distant member[3].

Thus each group authority transmits new key information with frequency T, and includes mechanisms for preventing forgeries. The system parameters, all of which are calculated by the group authority g, and refreshed every time period include:

- $s_g \in \mathbb{Z}_q$ - the "master" key for each group. Used to assign secret shares for symmetric key distribution.
- $S_i \in g$, the secret share for each member in g.
- Pu_g, Pr_g, the public/private key pair for the group authority.
- Pu_i, Pr_i, the public/private key pair for each host in the network, generated by the group authority.

Each host $X \in g$ is sent the following tuple (written on separate lines for clarity):

1. $Pu_g^{(new)}$, the new public key of g.
2. S_X, X, new secret share (elliptic curve element) used for pairing operations. New identity within g.
3. Pr_X, Pu_X, new public/private key pair for X.
4. $Enc_{Pr_g}(Pu_X)$, encrypted public key of X. Used for outgoing integrity signatures (to prevent X from generating its own public key).
5. $H(Pr_X^{(old)} \circ Pu_X^{(old)})$, a hash of the old (current) key pair. X only accepts the new key information if the sender of the tuple also knows its previous key pair. Thus, even if all current key information is lost, an attacker cannot inject new key information unless the previous set of key information has been compromised.

These fields are concatenated into one string, and encrypted first by $Pr_g^{(old)}$ using RSA encryption, then encrypted again using a symmetric key by K_{gX}, the symmetric key from the pairing described in section 3.1. Formally[4],

$$Enc_{KgX}\{Enc_{Pr_g}\{(Pu_g^{(new)}, S_X, X, Pr_X, Pu_X, Enc_{Pr_g^{(new)}}(Pu_X^{(new)}), H(Pr_X^{(old)} \circ Pu_X^{(old)}))\}\}$$

[2]The best known algorithms for solving the Discrete Logarithm problem over any cyclic group such as Pollard's Rho or Baby-Step-Giant-Step are on the order of the square root of the size of the group. The complexity bound for solving for the exponent should decrease linearly in the amount of cores dedicated to cracking it.

[3]Where $|g_i|$ is the size of the i^{th} group, we should expect the scheme to generate only $|g_i|$ bundles per time period T_{g_i}.

[4]A more computationally efficient encoding of this string may be to use Pr_g to encrypt a symmetric KEK, which then encrypts the rest of the contents.

5. CONCLUSION

This paper suggests a non-interactive key management scheme that may be deployed in challenged networks using the Bundle Security Protocol. It is defined in terms of well-studied cryptographic primitives, namely the NIST-certified HMAC-SHA1 and PBKDF2 functions, and is ultimately reducible to the difficulty solving the Discrete Logarithm problem over an elliptic curve of prime order. It assumes some key information is pre-distributed, but many networks of sufficient connectivity and density could bootstrap deployed hosts with the use of a perfectly-secure secret sharing scheme. Furthermore, it maintains the PIB requirement that integrity signatures can be verified en-route to the security destination.

The essential aspect is that all key-related information is available up-front, and it decouples the act of verifying a signature from querying the key authority. The identity-based aspect allows for the construction of distinct pairwise secrets and implicit authentication to members of a common group who have just become aware of the other's existence[5].

6. REFERENCES

[1] Aniket Kate and Greg Zaverucha and Urs Hengartner. "Anonymity and Decurity in Delay Tolerant Networks". Technical report, In SecureComm, 2007.

[2] Paulo S. L. M. Barreto. "the pairing based crypto lounge". http://www.larc.usp.br/ pbarreto/pblounge.html.

[3] Dan Boneh and Matthew Franklin. "Identity-Based Encryption from the Weil Pairing". *SIAM J. Comput.*, 32(3):586–615, 2003.

[4] Règis Dupont and Andreas Enge. "Practical Non-Interactive Key Distribution Based on Pairings". In *Proceedings of the International Workshop on Coding and Cryptography (WCC)*, 2002.

[5] Jeremy Horwitz and Ben Lynn. "Toward Hierarchical Identity-Based Encryption", 2002.

[6] John Bethencourt, Carnegie Mellon University. "Intro to Bilinear Maps (Lecture Notes)".

[7] Liu, Fang and Cheng, Xiuzhen. "LKE: A Self-Configuring Scheme for Location-Aware Key Establishment in Wireless Sensor Networks". *IEEE Transactions on Wireless Communications*, 7(1):224–232, 2008.

[8] Liu, Fang and Cheng, Xiuzhen and Ma, Liran and Xing, Kai. "SBK: A Self-Configuring Framework for Bootstrapping Keys in Sensor Networks". *IEEE Transactions on Mobile Computing*, 7(7), 2008.

[9] Srdjan Čapkun and Levente Buttyán and Jean-Pierre Hubaux. "Self-Organized Public-Key Management for Mobile Ad Hoc Networks". *IEEE Transactions on Mobile Computing*, 2:52–64, 2003.

[5]There are other security benefits as well: In contrast to a password-based scheme using a group key that may enable the same functionality, the compromise of a single host in the identity-based scheme would keep all other pairwise secrets in the group reasonable security (reducible to DL). Without the exposure of a host's secret, S_X, an attacker must capture an authenticated message and then compute $KDF(e(A,B)^i, n)$ for each $i \in \mathbb{Z}_q$ to attempt to brute force s.