Key Distribution Mechanism for Symmetric Cipher using Smart-Its Friends Concept

Telecooperation Office

Proposal for Praktikum Thema

May-Jun 2002

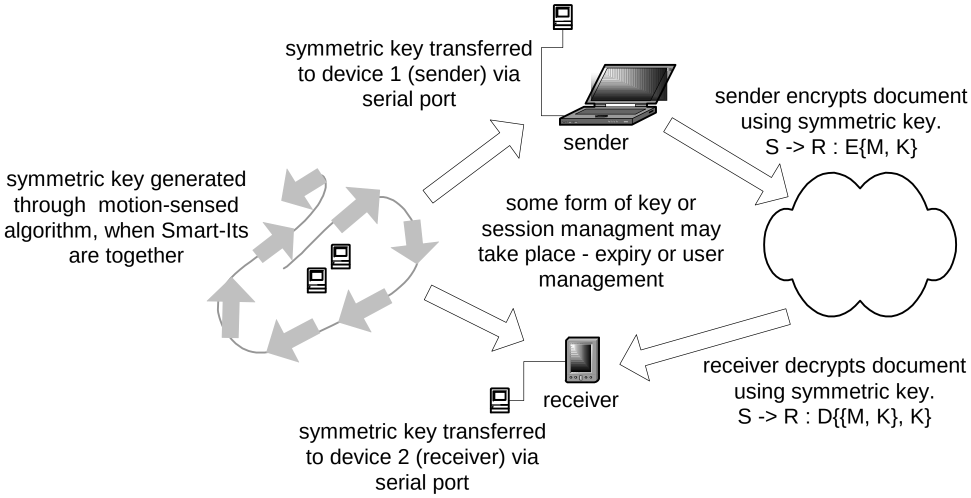
Keywords. Symmetric Cryptography, Smart-Its, Symmetric key distribution, Advanced Encryption Standard (AES), JCE (Java Cryptography Extension),

Background

Cryptography is the means by which the integrity of electronic data is protected as it is transmitted from sender to receiver. Integrity implies that it successfully arrives at the intended recipient, its contents are unaltered, an illegitimate viewer has not read it during transmission, and the receiver can guarantee the source of the data. Symmetric (or secret key) cryptographic algorithms (*3DES, IDEA, AES*) are the classical form of cryptography, where sender and receiver share the same secret key upon which the data encoding/ encrypting is based. The short fall of symmetric algorithms is the challenge of securely exchanging the secret key, as this had to be done using plaintext. This was the motivation for more advanced key distribution mechanisms, which led to the development of public key algorithms (most popular are *Diffe-Hellman* and *RSA*). Public key algorithms are based on an encryption function that requires one key for the encryption (the public key) and another for decryption (the private key). The encryption key or public key can therefore be freely distributed as only the owner of its corresponding decryption or private key may decode an encrypted message. However, these algorithms consume relatively high amounts of computational resources, yet the strength of the cipher is significantly weaker than a symmetric one. Nevertheless, secure key distribution is the first concern before thinking about low-probability attacks on a cipher.

Smart-Its friends implement a simple interaction concept that augments the association of objects based on the context of location proximity. The semantics of this association are based on recognition of unique and synchronous turning points observed by the associated objects (consider hand shaking). The uniqueness of the binary representation of this event and the storage capacity of the Smart-Its are sufficient to generate an acceptably resilient key length (128 bits or more) with which a robust symmetric algorithm (such as AES) could be computed. The initiative is not for the Smart-Its to act as encryption modules, but rather as the key generation and implicit key exchange mechanism. This is aimed at countering the key distribution issue supporting the usage of a symmetric cipher. As hinted in the previous paragraph, being capable of using a symmetric algorithm with a secure secret key distribution mechanism is desirable for small computational power devices such as mobile phones and PDA's. For example, algorithms like RSA are totally infeasible to be computed on today's mobile phones! The further work required is to transfer the generated key (or seed) from the Smart-Its to the device hosting the applications involved in the secure information transfer.

Methodology and Tasks


symmetric key transferred to device 1 (sender) via serial port
sender
sender encrypts document using symmetric key. S -> R : E{M, K}
symmetric key generated through motion-sensed algorithm, when Smart-Its are together
some form of key or session managment may take place - expiry or user management
receiver
receiver decrypts document using symmetric key. S -> R : D{{M, K}, K}
symmetric key transferred to device 2 (receiver) via serial port

1) Define and implement the key-generation algorithm using the Smart-Its friends technique
2) Implement a small application (suggest using Java and JCE) that selects a file to be encrypted/ decrypted (suggest using the AES) and communicates with the Smart-It over the serial port to obtain the key for encryption/ decryption.
3) Specify methods for evaluating the mechanisms
4) Propose possible security threats and formulate attacks based on these threats
5) Suggest enhancements and issues for key management

[Philip Robinson. Philip@teco.edu]