Evaluation of real-time aspects of multiparty security on low-power mobile devices

Tobias Limmer[⋆], Falko Dressler[⋆], Ruben Gonzalez[†]

[⋆] University of Erlangen
Department of Computer Science

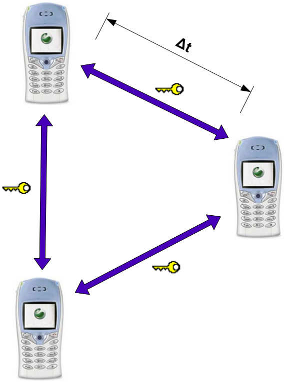
[†] Griffith University
Gold Coast Campus, Australia

1.12.2006

Overview

Problem description

- we want to develop an networked application with real-time properties (e.g. video conferencing)
    - on mobile phones
    - with multiparty communication
    - using secure data transfer
        - eavesdropping must not be possible
    - ad-hoc network, no third party server available
- is it possible? At what performance?



Hardware of mobile phones

- slower generation: series 40 mobile phones
  - processor: 32bit ARM-9 with 20 MHz
  - RAM: 200 KB
    - 70 KB available for applications
- faster generation: series 60 mobile phones
  - processor: 32bit ARM-9 with 120 MHz
  - RAM: 2 MB and more
  - growing market share, caused by falling prices
- even faster: PDA-like devices
  - not included in my work
- mobile networks (speed achieved in current networks): GSM (9.6 kbit/s), GPRS (25 kbit/s), UMTS (up to 384 kbit/s), UMTS with HSDPA (1.8MBit/s)

Software on mobile phones

- program environment: Java ME (Java 2 Micro Edition)
- programs run on different architectures without modifications[1]
- available protocols for communication: TCP/IP, UDP/IP (and several protocols in the application layer)
- performance disadvantages:
  - application can not access features of real-time operating system:
    - primary task is to maintain network connection (GSM/UMTS)
    - custom Java applications get "idle" time
  - programs run in virtual machine
  - no direct access to hardware → no hardware based optimization possible

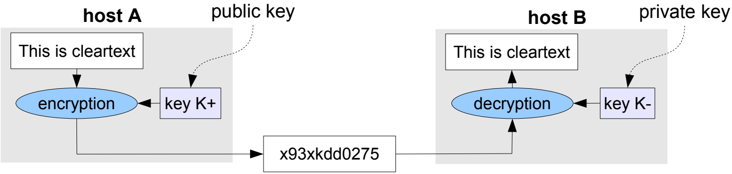
[1] in theory

Types of cryptographic algorithms

- Symmetric key algorithms
  - both parties use same key for encryption and decryption
  - $\rightarrow$ key exchange more difficult
  - low computational costs
    - used data operations: bit permutation, XOR, addition, ...
    - small key sizes considered secure (128 bit)
  - used for encryption of application data
  - existing algorithms: AES, 3DES, Blowfish, ...

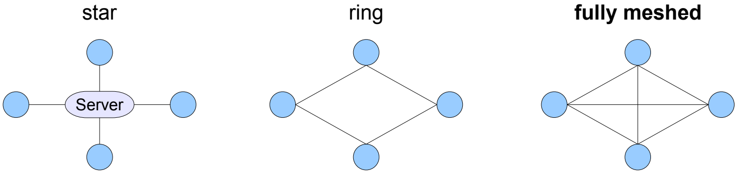
Types of cryptographic algorithms

- Asymmetric algorithms / public-key algorithms
  - each party generates a public and a private key which are mathematically related
  - data encrypted with public key $K+$ can only be decrypted with corresponding private key $K-$ and vice versa
  - every host knows or can retrieve other hosts' public keys
  - private key is very hard to deduce from public key
  - public key and some additional information is called certificate
    - enables mutual authentication



Multiparty communication

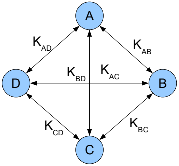
- simultaneous communication among several peers
- requirement: peers join and leave dynamically
- needed for applications like
  - teleconferencing
  - groupware applications (e.g. shared white boards)
  - games
- available network topologies:



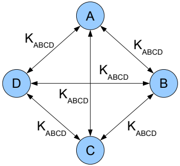
- used topology: fully meshed net → each device is connected to all other devices
  - → identical data is sent multiple times to all other hosts in the group

Sharing encryption keys

- possible connection management types:
  - separately managed connections
    - each host establishes secure connections with all other hosts
    - data has to be encrypted several times with different keys
      → memory overhead, high computational cost
  - group communication protocol
    - all hosts share common symmetric encryption key
    - data only needs to be encrypted one time
  - keys have to be distributed by a "key agreement protocol"



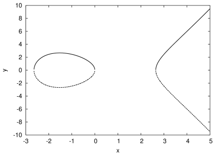
separate encryption keys



shared encryption keys

Test of asymmetric cryptographic algorithms

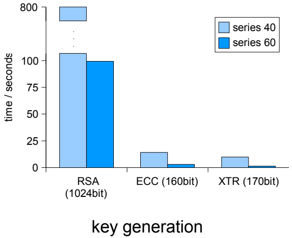
- key agreement phase of secure protocols and authentication systems usually use asymmetric cryptographic algorithms
- several algorithms benchmarked:
  - RSA (by Rivest, Shamir, Adleman in 1978)
    - based on the discrete logarithm problem
  - ECC (by Miller in 1985 and Koblitz in 1987)
    - based on mathematics of finite field elliptic curves
  - XTR (by Lenstra and Verheul in 2000)
    - improves RSA by using traces to represent and calculate powers

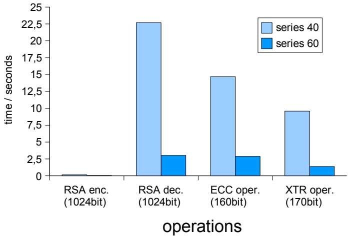


example of an elliptic curve

Results of test

- compared speeds on series 40 and series 60 mobile phones
- key sizes with similar security level:
  RSA (1024 bit), ECC (160 bit), XTR (170 bit)

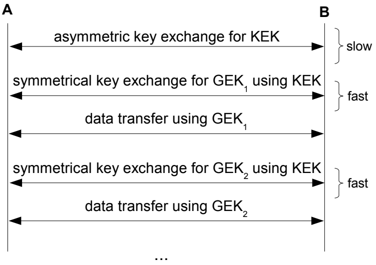




- Results:
  - ECC is faster than RSA, but XTR is even faster
  - XTR can not be used, as
    - no secure protocols with authentication are available
    - it is not considered mature (thus it is not considered secure either)

Key agreement protocols

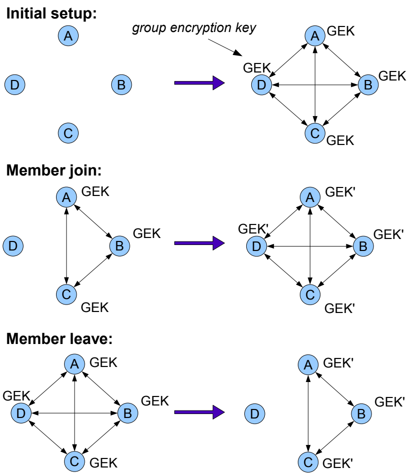
- desired tasks
  - performs authentication of all members
  - distributes common encryption key
- makes use of asymmetric cryptography and is most important phase of the secure communication's setup
- KEK: key encryption key which is used to distribute GEK (group encryption key) with symmetric encryption techniques
- GEK is changed regularly, KEK is only changed when members join or leave group



Key agreement protocol analysis

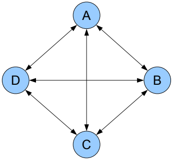
- theoretical analysis of protocols' performance
- only sequential delays are included
- conditions:
  - all members are realized on identical (or similar) devices
  - no third party, like a server
- only network lag and asymmetrical operations are considered
  - other operations like symmetrical encryption and decryption, random number generation, etc. are very fast and not noticeable during execution

member operations:



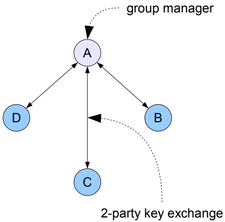
Types of key agreement protocols

- contributory key agreement protocols
  - all hosts are treated identically
  - advantages:
    - high security: all hosts take part in calculation of encryption key (randomness of hosts is combined)
    - computational load is spread evenly between all group members
  - disadvantages:
    - all group members need to perform expensive asymmetric calculations for all operations
  - tested protocols:
    - AKE1 by Bresson in 2001 (ring-based)
    - TGDH by Kim in 2000 (tree-based, modified to provide authentication)



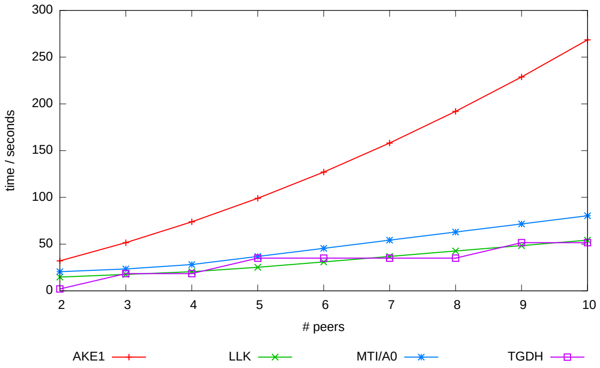
Types of key agreement protocols

- distributed/centralized key exchange protocols
  - one central host called group manager (GM) coordinates key exchange and authenticates all members
  - 2-party key exchange protocols are used between GM and members
  - advantages:
    - member leave operations do not need any asymmetrical calculations
  - disadvantages:
    - computational load is mainly on GM → which becomes bottleneck
    - members have to trust GM
  - tested protocols: LLK by Lee in 1998 and MTI/A0 by Matsumoto in 1986



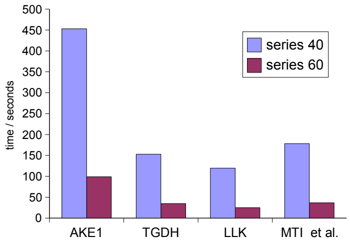
Results of analysis

- theoretical results on series 60 devices for initial setup:
  - used ECC with 160bit keys
  - protocol complexities (n: group size):
    - AKE1: $O(n^2)$, TGDH: $O(\log n)$, LLK $O(n)$ and MTI/A0 $O(n)$

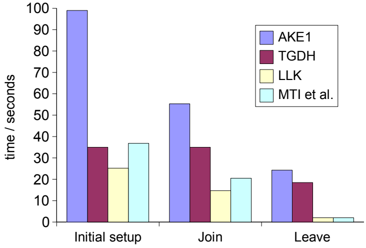


Results of analysis

- results for group with 5 members:



Initial setup compared between S40 and S60

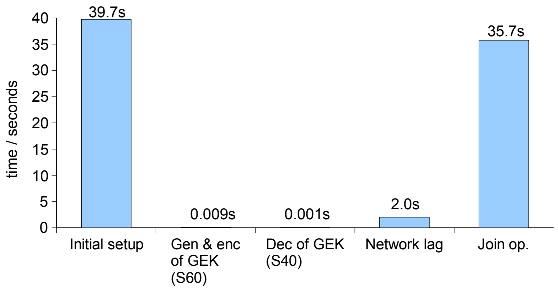


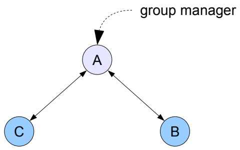
Protocol performance on S60 devices

- series 40 mobile phones too slow for application
- contributory protocols AKE1 and TGDH in most cases slower than centralized protocols LLK and MTI/A0
- centralized protocol with LLK performs best

Benchmark for key exchange protocol

- real world test of centralized key exchange protocol using LLK
- setup:
  - 3 devices:
    - series 60 device as group manager
    - two series 40 devices as group members
  - GSM network





Conclusion

- series 40 mobile phones with 20 MHz too slow - initial setup of a secure group takes at least 2 minutes for 5 members
- key exchange in a group with 5 members with series 60 mobile phones takes 25.2 seconds
  - only appropriate for few applications
  - use of trusted third party would reduce execution time significantly
  - speed of mobile phones increases rapidly (MP3, video streaming, ...), just wait a few years?
- symmetric encryption generally possible
  - by use of pre-shared keys no asymmetric algorithms are needed

The end

Thanks for your attention!

Questions?