Building Survivable Services using Redundancy and Adaptation

Rick Schlichting

AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07974, USA

Work done in collaboration with Matti Hiltunen (AT&T) and Carlos Ugarte (Univ. of Arizona).

Supported at Arizona in part by DARPA under grant N66001-97-C-8518 and NSF under grant ANI-9979438.

Terminology Macro

declare survivable **macro**

if (JCLaprie | disciple)
  then macrodef(survivable, "dependable")
else if (FBSchneider | BGates | disciple)
  then macrodef(survivable, "trustworthy")
else if (DARPA | disciple | PI)
  then macrodef(survivable, "robust")
else if (CISCO | disciple)
  then macrodef(survivable, "resilient")
else macrodef(survivable, "survivable")

Introduction

Survivable systems continue providing their service despite failures and intrusions.

Survivable services designed to provide core functionality for survivability in networked systems.

➡ Focus on using redundancy and adaptation to implement survivable services.

Themes/caveats:

- Explore the use of traditional fault-tolerance techniques in this context.
- Focus largely on system structuring and mechanisms, not policies.
- Used in combination with other techniques.
- Not much on assurance.

Outline

- Redundancy and adaptation
- System support and Cactus
- Example: survivable SecComm
- Related work
- Conclusions

AT&T

Redundancy

Traditional fault tolerance:

- Time redundancy: repeated execution, retransmission.
- Space redundancy: replication of data/computation.

Both can be (and have been) used to increase survivability.

Redundant methods: Use two or more methods to enforce a security property.

Goal: Properties ensured through redundancy should remain valid even if some of the methods used have been compromised.

Example: Confidentiality in communication security.

- Successive encryption with different methods.
- Alternating order of methods used.
- Apply different methods to different messages in a stream.

Example: Authentication.

- Two or more independent authentication services (e.g., PKI, Kerberos).
- Multiple user authentication methods (password, biometrics).

Impact of redundancy:

- Eliminates single points of vulnerability
- Introduces artificial diversity into the system
- Introduces unpredictability

AT&T

Role of Independence

Redundancy increases survivability only if methods are independent, i.e., breaking one does not make it easier to break others.

Analogous to failure independence in fault tolerance.

Example: Sources of dependency in communication security:

- Same key used by different methods.
- Same key creation/distribution method used.
- Keys stored in the same place.
- Methods of combining encryption algorithms.
- Etc.

AT&T

Techniques to increase independence:

- Use different keys established using different key distribution methods (e.g., Diffie-Hellman and Kerberos).
- Unrelated encryption methods, e.g., different block sizes.
- Combination techniques that increase independence.

Redundancy can also be used for integrity, i.e., multiple message signatures.

Redundant methods in other services:

- Redundancy for PKI and certification agencies.
- Redundancy in file access control:

  Encrypted files (user must be both authorized and have the key),

  Monitoring for changes to important files (e.g., web pages, log files).
- IDS viewed as a redundant "failure detection" service.

AT&T

Adaptation

Adaptation: Changing execution behavior dynamically.

Two types:

- Value adaptations and algorithmic adaptations.
- Changing parameters vs. changing methods.
- Both useful for survivability:

Predictive: Adapt methods when attack anticipated.

Reactive: Adapt compromised methods if an attack detected (e.g., IDS.

Preventive: Adapt methods and parameters non-deterministically at runtime to increase artificial diversity and unpredictability.

Impact:

- Introduces artificial diversity into the system
- Introduces unpredictability
- Provides an approach for dealing with detected intrusion attempts.
- Provides an approach for graceful degradation
- Provides an approach for dealing with changes in user security requirements.

Caveat:

- Adaptation mechanisms must not make the service more vulnerable by introducing new attack modes.

System Support

Issue: What kind of system support needed to build survivable services based on redundancy and adaptation?

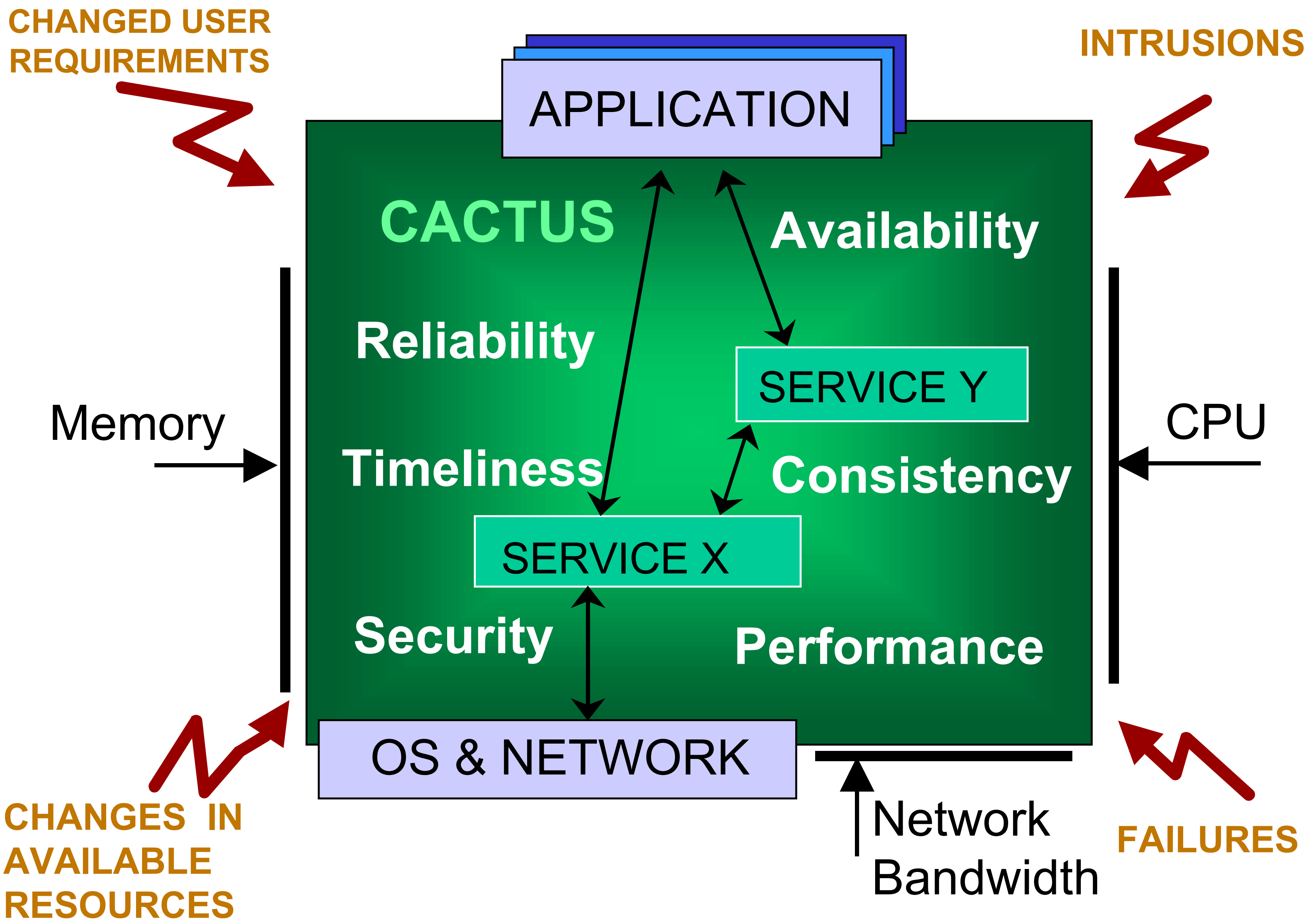
Our answer: a software customization framework.

Cactus:

- Supports construction of configurable services and protocols in networked systems.
- Configurability ⇨ multiple redundant methods.
- Dynamic ⇨ adaptive reactions.
- System supports coordinated value and algorithmic adaptations.

AT&T

Cactus Vision



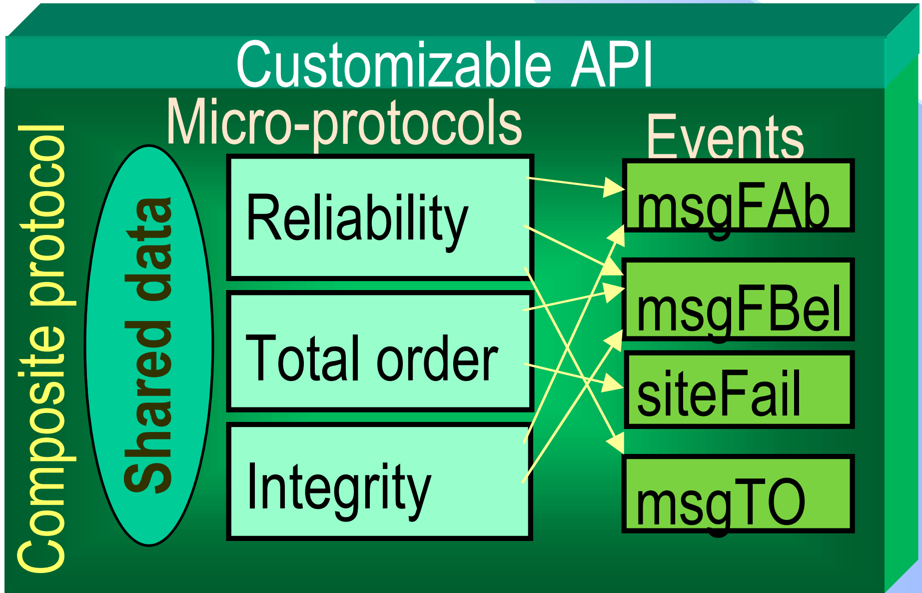
Cactus Approach

A protocol/service implemented as a composite protocol composed of micro-protocols - each implements a function or property.

Service customized by configuring the service with the appropriate micro-protocols.

Cactus mechanisms support configurability:

- Flexible event mechanism.
- Shared data.
- Dynamic messages.



Example: SecComm

SecComm: customizable secure communication service implemented using Cactus.

- Basic security MPs for privacy, integrity, authenticity, non-repudiation, replay prevention, key distribution, ….
- Implement well-known security algorithms such as DES, RSA, IDEA, MD5, SHA, etc.
- Key distribution MPs provide keys to basic security MPs as needed; allow keys to be chosen by one or both principals, or by a third party.
- MPs simple ⇒ easy to add custom security MPs.

Secure but not Survivable

SecComm service not survivable.

Multiple single points of vulnerability; security compromised if

- Key stolen,
- Encryption method broken, or
- Key distribution method/service broken.

Traditional solution: increase key length or use a stronger cryptographic method.

Adequate for survivability?

Using Redundancy

Goal: Security property should remain valid even if some methods compromised.

Example: For confidentiality, possible approaches:

- Successive encryption with different methods.
- Alternating order of methods used.
- Apply different methods to different messages in a stream.

Result: Breaking one method/key not enough to compromise security completely.

AT&T

Using Adaptation

Goal: Change methods using predictive, reactive, or preventive adaptation.

Example: For confidentiality, possible adaptations:

- Coordinated key change
- Coordinated switching of encryption MPs
- Coordinated activation of additional (redundant) encryption MPs
- Coordinated deactivation of redundant encryption MPs.

Result: Replace compromised methods at runtime.

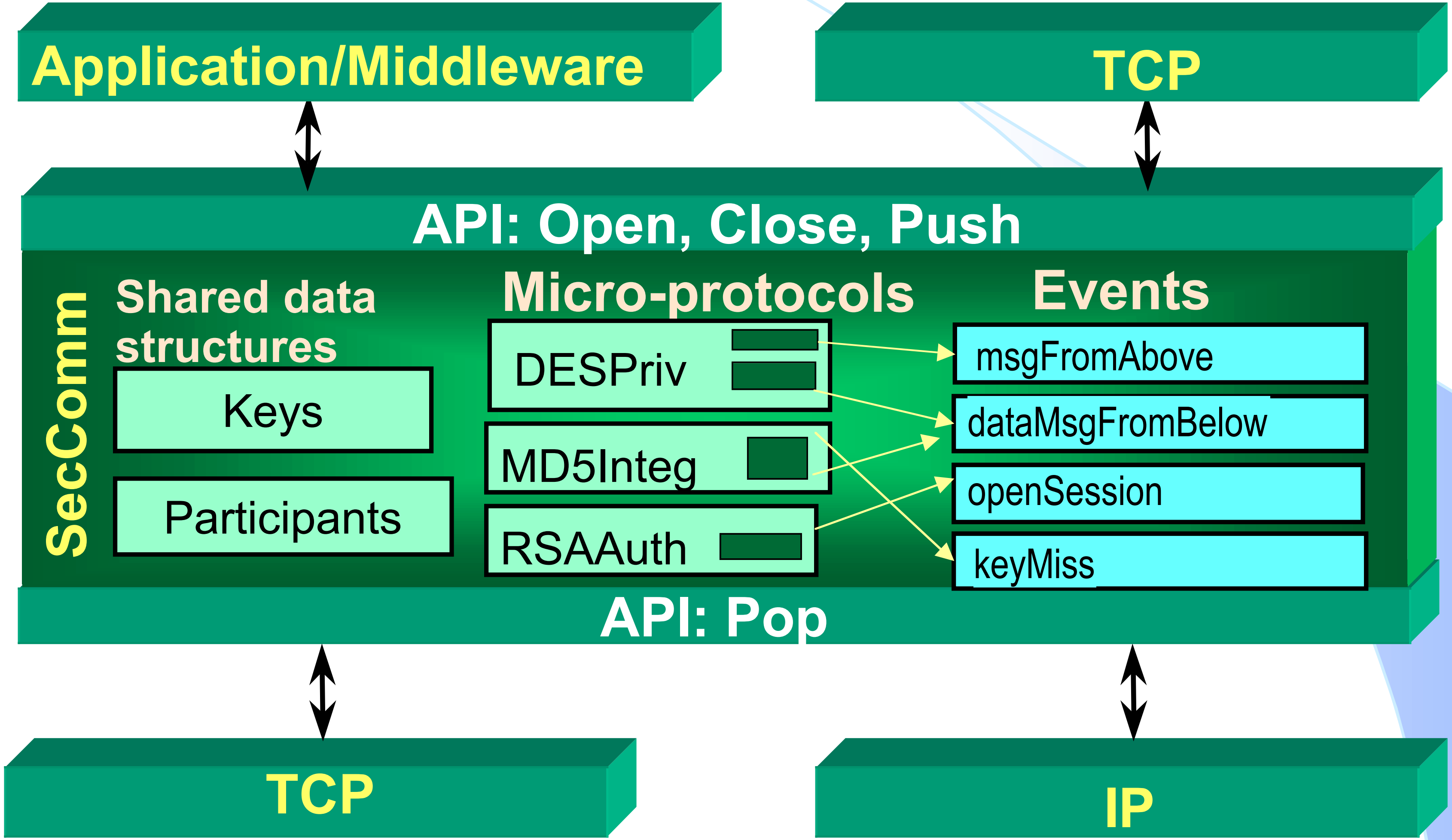
Survivable SecComm

SecComm with MPs that support redundancy and adaptation.

- Redundancy: meta security MPs, construct more complex security protocols using basic security MPs, e.g., multiple encryption, alternating encryption.
- Arbitrary number and combinations of the MPs possible.
- Adaptation: Adaptation MPs, coordinated swapping of basic and meta security MPs using various adaptation protocols.

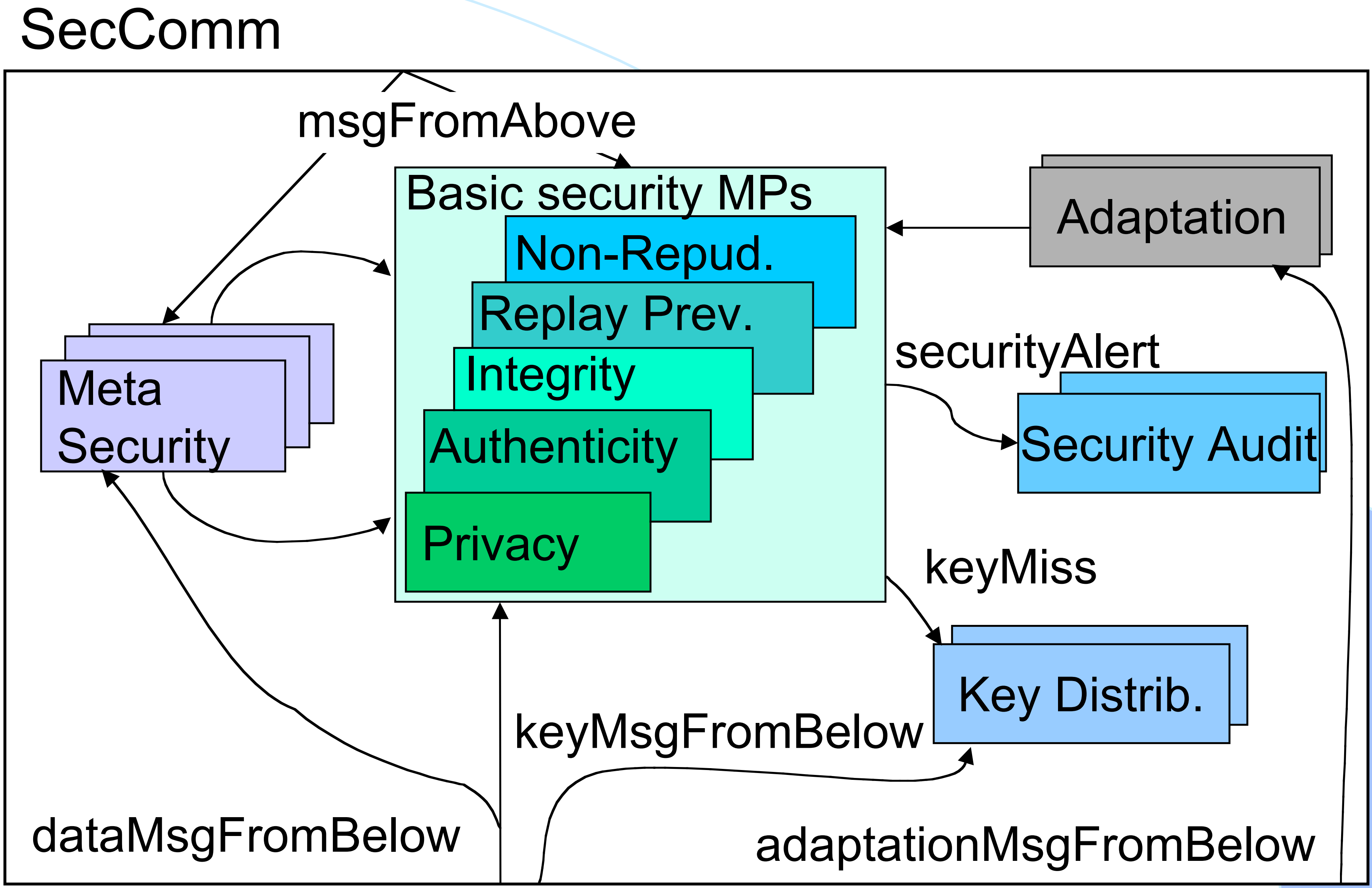
AT&T

SecComm in Cactus



AT&T

MP classes and event interactions



Basic Security MPs

Implement basic transformations: encryption, signatures, etc.

```
micro-protocol BasicSecurity(dEvnt,dOrd, uEvnt, uOrd, key){
    handler ProcessDownMsg(msg){
        if Keys[myKey] == NULL raise(keyMiss,myKey,SYNC);
        add attributes, pack, encrypt, etc;}
    handler ProcessUpMsg(msg){ … }
    initial { myKey = key; bind(dEvnt,ProcessDownMsg,dOrd);
        bind(uEvnt,ProcessUpMsg,uOrd);} }
```

dEvnt and uEvnt are pointers to Cactus events that may be the events msgFromAbove and dataMsgFromBelow or some events raised by meta security MPs.

Meta Security MPs

Construct more complex security protocols out of basic MPs, e.g., redundancy and alternation techniques.

```
micro-protocol MetaSecurity(dEvnt,dOrd, uEvnt, uOrd,
                              dBasicEvnts, uBasicEvnts){
   handler ProcessDownMsg(msg){
        in some order raise(dBasicEvnts[i],msg,SYNC); }
   handler ProcessUpMsg(msg){ … }
   initial { bind(dEvnt,ProcessDownMsg,dOrd);
        bind(uEvnt,ProcessUpMsg,uOrd);} }
```

dBasicEvnts and uBasicEvnts are vectors of pointers to Cactus events.

Adaptation MPs

Coordinate the swapping of basic and meta security MPs at runtime.

micro-protocol SimpleAdaptation(…){
handler StartMaster(...){ *deactivate old MP for outgoing messages; send "adaptation start msg" to slave;* }
handler StartSlave(...){ *send "adaptation ack msg" to master; deactivate old mp; activate new mp;* }
handler SwitchMaster(...){ *deactivate old mp for incoming messages; activate new mp;* }
initial { … }}

This adaptation MP is asymmetric; symmetric MPs also exist.

SecComm Performance

Test environment:

- Cactus/C 2.2 on Linux.
- 600 MHz Pentium III PCs.
- Linux 2.4.7.
- 1 Gbit Ethernet.

Testing method:

- 100-byte messages.
- average roundtrip times over > 1000 roundtrips.

Key sizes and modes:

- DES: 56-bit key in CFB mode.
- Blowfish: 448-bit key in CFB mode.
- IDEA: 128-bit key in CFB mode.
- XOR: 64-bit "key".

AT&T

Roundtrip times in μs.

286 + 326 = 612 > 549

Configuration	RTT	C/O IP	C/O Base
IP	365	n/a	n/a
Base SecComm	409	44	n/a
DESPrivacy	695	330	286
BlowfishPriv.	659	294	250
MD5Integrity	735	370	326
DES + MD5	958	593	549
MultiSec: DES + Blowfish	892	527	483
+ XOR	996	631	587
AltSec: DES + Blowfish	712	347	303
+ XOR	700	335	291

DES + Blowfish:
286 + 250 = 536 > 483

AT&T

Related Work

Redundancy techniques:

- File systems/data storage: encryption, fragmentation/repl.
- Detection: Tripwire, StackGuard, IDSs.

Adaptation techniques:

- ITUA: unpredictable adaptations in GC system.
- Ensemble: swap one protocol stack for another.

Secure communication:

- IPSec, SSL/TLS: Some choice of methods, limited support for redundant methods.

Configuration frameworks:

- x-kernel, Ensemble: general hierarchical composition frameworks used for security.
- Antigone: configuration framework for security policies in GC.

Conclusions

Thesis: Redundancy and adaptation techniques can be used to increase the survivability of services.

Independence is a key requirement.

Cactus and SecComm demonstrate system support for redundancy and adaptation techniques.

Configurability in general can be viewed as a method to create artificial diversity and increase unpredictability.

Future work: Developing more adaptation protocols and making the adaptation itself more survivable.

AT&T