



Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks

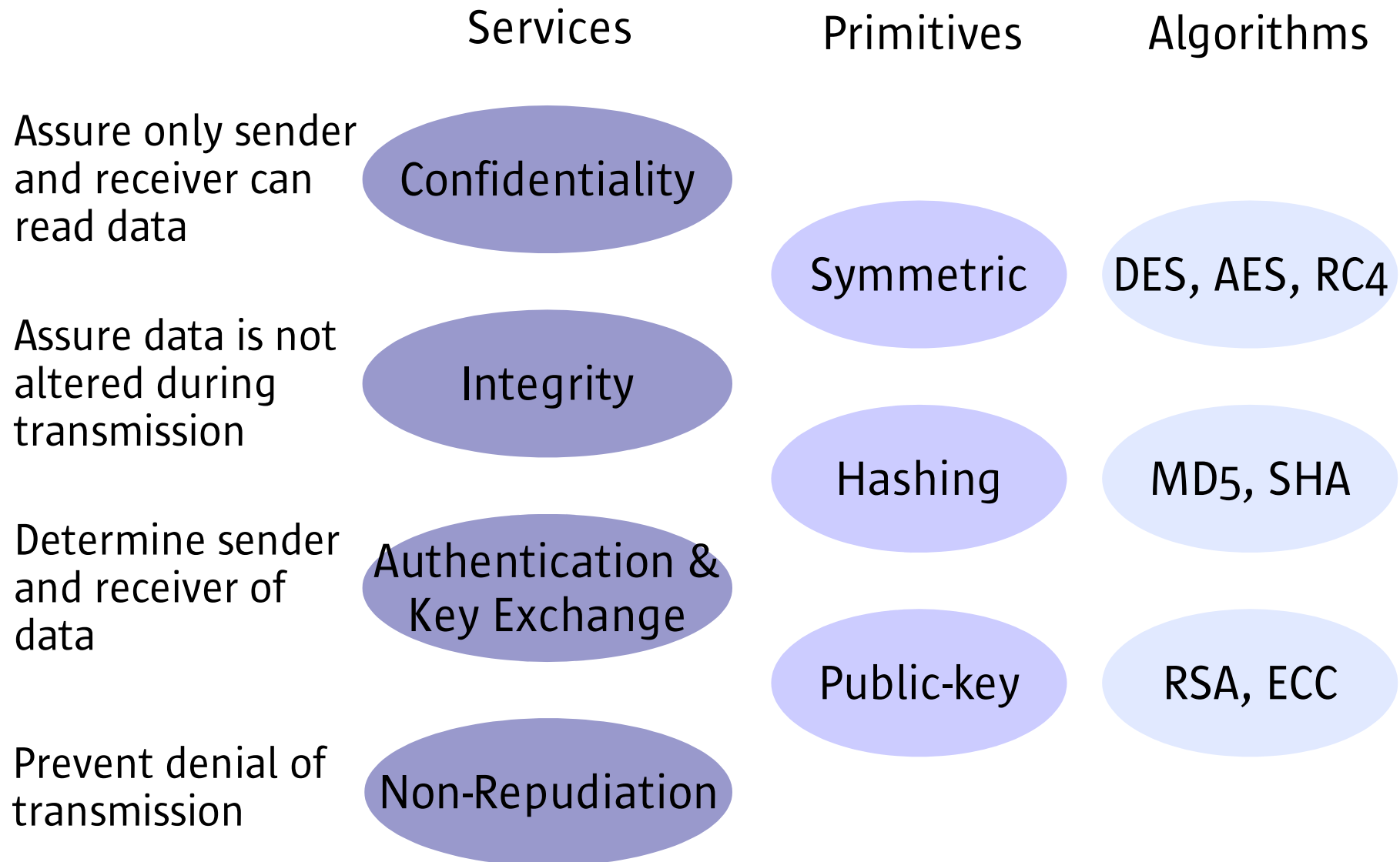
Arvinderpal Wander

Nils Gura, Hans Eberle, Vipul Gupta,
Sheueling Chang

Sun Microsystems
Laboratories



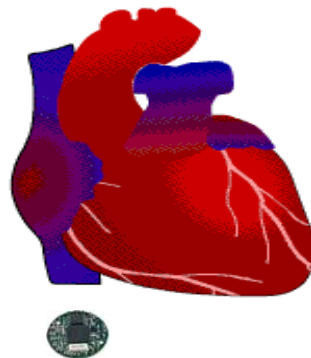
Security Services/Primitives/Algorithms



Security in Sensor Networks

- Why public-key ?
 - Sensor networks and ubiquitous computing with wireless communication
 - Need for mutual authentication
 - Flexible key distribution
- Why software?
 - Cost
 - Usability on existing devices
 - Implementation flexibility

Building Automation Medical Devices



Industrial Automation



Surveillance Habitat Monitoring



Public-Key Algorithms

- RSA
 - Relies on integer modular exponentiation:
 - $[1024\text{-bit (17-bit/1024-bit)}] \bmod 1024\text{-bit}$
 - Value of the exponent = User's **public key**/**private key**
 - Most widely used public-key algorithm today
- ECC
 - Relies on scalar point multiplication:
 - $Q(X_2, Y_2) = (160\text{-bit scalar}) * P(X_1, Y_1)$
 - P & Q correspond to points on an elliptic curve
 - Scalar = User's **private key**, EC Point $Q(X_2, Y_2) =$ **public key**

Primitive RSA/ECC Operations

- Public-Key cryptography primitives provide
 - Encryption/Decryption
 - Signature/Verification
 - Key Exchange
- RSA
 - Signature: Modular exponentiation using private key
 - Verify: Modular exponentiation using public key
- ECC
 - Signature/Verification: ECDSA
 - Key Exchange: ECDH

Crossbow/Berkeley Motes Platform

- Processor specifications:
 - Atmel ATmega128L 8-bit CPU, 4MHz
 - 128KB code and 4KB data memory
- Wireless transceiver specifications:
 - Chipcon CC1000, 915MHz, 3mW
 - 12.4Kbps
- Transmitting 1 bit \approx 2,000 cycles of computation
- Transmit cost approximately 2x receive cost

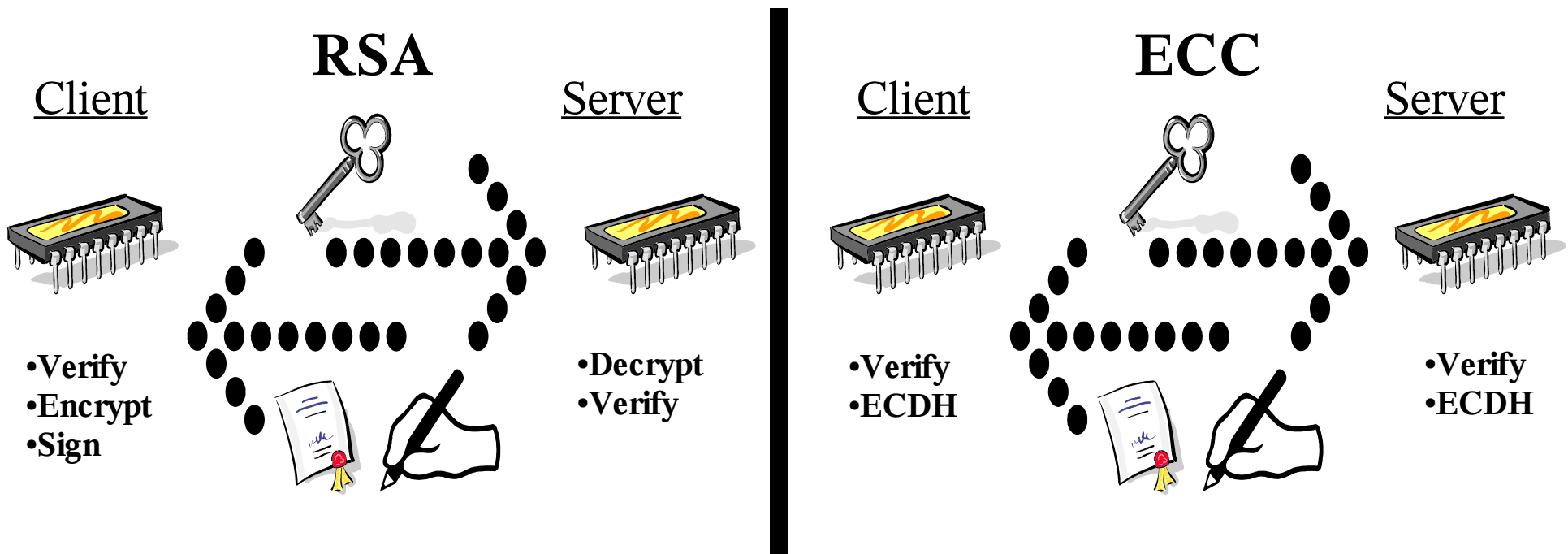
Energy Cost of Primitive Operations

Algorithm	Signature		Key Exchange		[mJ]
	Sign	Verify	Client	Server	
RSA-1024	304	11.9	15.4	304	
ECC-160	22.82	45.09	22.3	22.3	

- Sign [RSA/ECC] = Transmit [5,000B/400B]
- Sign [RSA/ECC] = AES-Encrypt[190KB/14KB]

RSA/ECC Handshakes

- Handshake
 - Authentication and key exchange
- Digital Certificates
 - Use only essential fields: ID, public key, CA signature

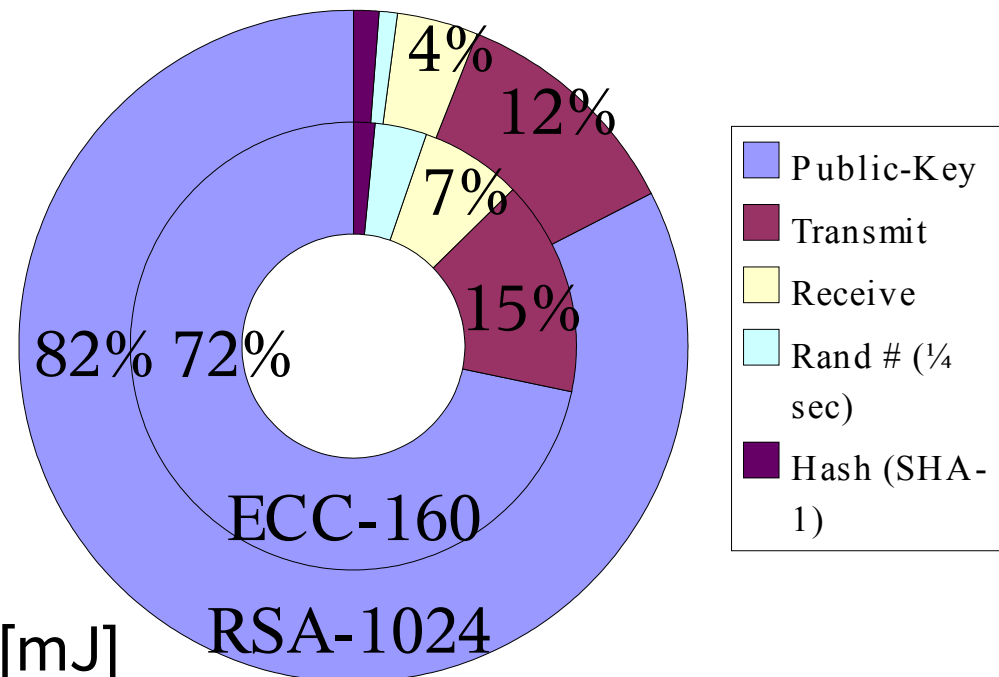


Cost Breakdown - Handshake

- RSA: Client/Server transmit 490/314-bytes
- ECC: Client/Server transmit 138/138-bytes
- RSA handshake:
 - Sign dominates
 - Requires **4x** energy of ECC handshake

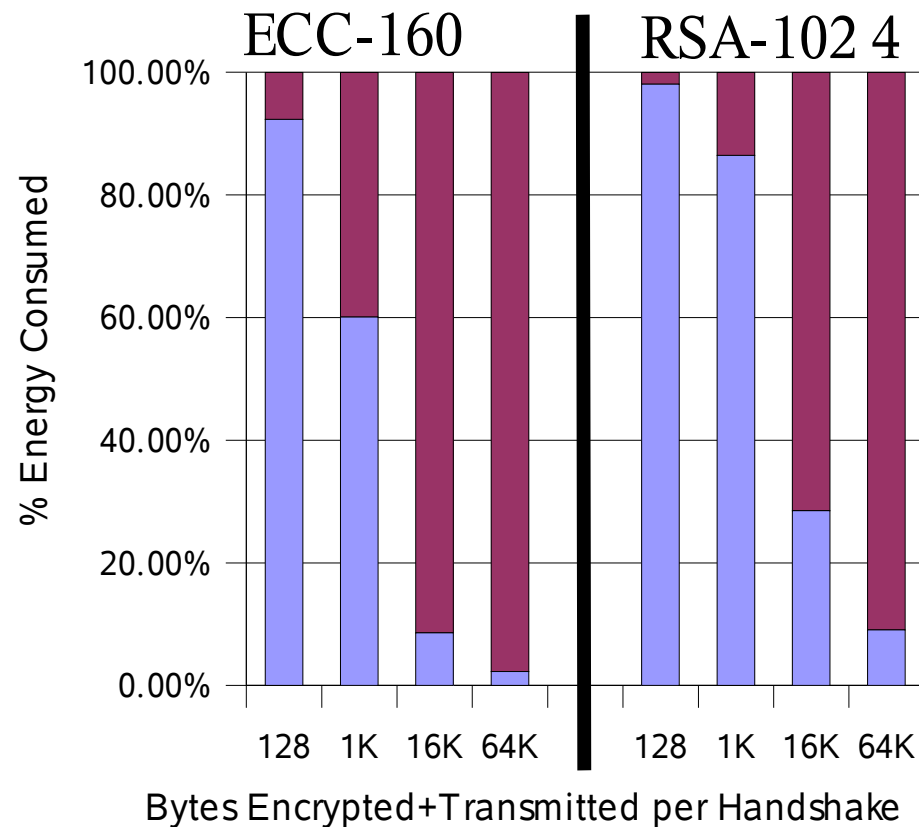
	Client	Server
RSA-1024	397.7	390.3
ECC-160	93.7	93.9

[mJ]



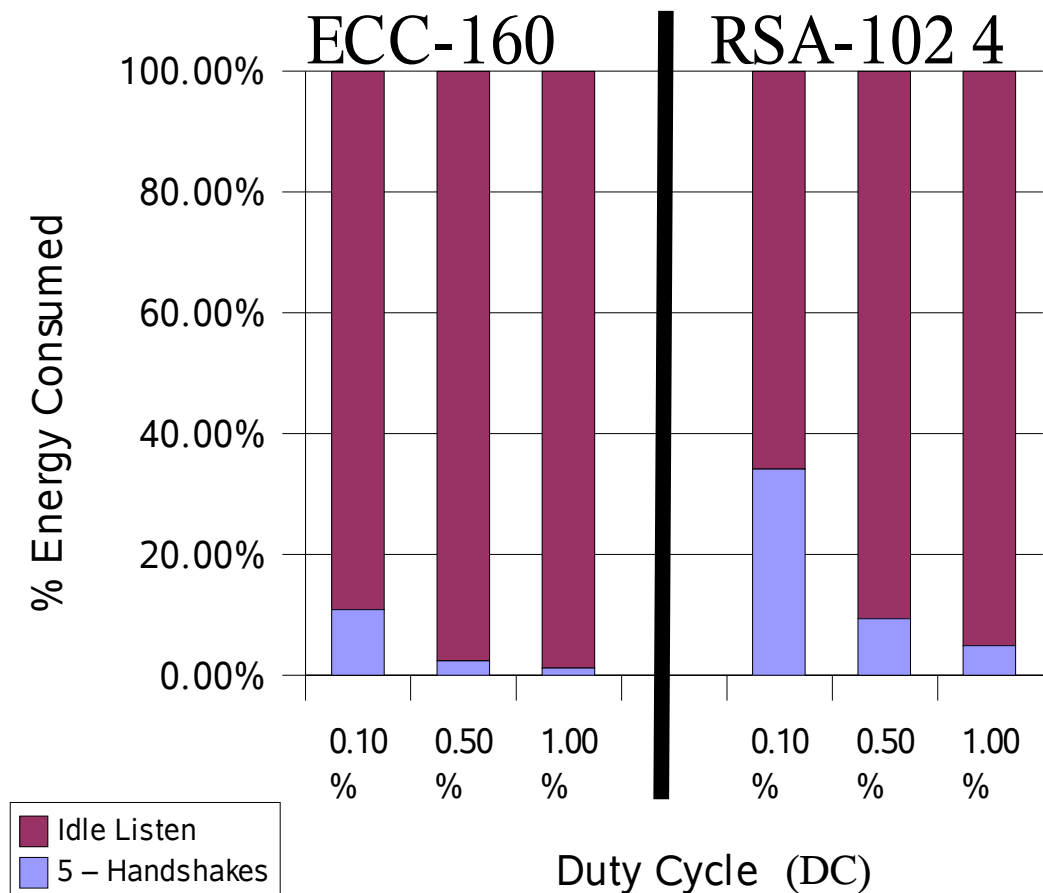
Long-Term Cost Comparison

- ECC:
 - < 10% after 16KB
- RSA:
 - <10% after 64KB
- Encryption:
 - AES-128
- Transmit Power:
 - 3mW (max)



Long-Term Cost Comparison

- 5 handshakes/day
- Cost approaches zero as DC increased
- ECC
 - <11%
 - 1.21% at DC=1%
- Note:
 - Ignores all other costs



Conclusions

- Software-based public-key implementations viable
 - Cost nearly negligible if handshakes are infrequent
- ECC performance better due to smaller keys and smaller communication overhead
- Reasonable energy consumption and processing time
 - ECC point mul – 1.61 sec versus RSA mod exp – 22 sec
- Future work
 - Group key exchange protocols with minimal public-key handshakes