Mobile Banking Security Algorithms Based on DES and RSA

Ravi Reddy
Independent Researcher
India

ABSTRACT

Mobile banking has revolutionized financial services by enabling transactions anywhere, anytime. However, security remains a critical concern, especially on resource-constrained mobile devices. This manuscript presents an in-depth study of two classic symmetric-asymmetric hybrid schemes—Data Encryption Standard (DES) for bulk data encryption and Rivest–Shamir–Adleman (RSA) for secure key exchange—within the constraints of pre-2016 mobile platforms. We analyze their performance, resilience to common attacks (e.g., known-plaintext, chosen-ciphertext), and usability on devices typical of 2015. Statistical analysis quantifies encryption/decryption latency and throughput. Simulation experiments using NS-2 model session setup, key exchange, and transaction workflows. Results indicate that DES+RSA hybrids achieve acceptable security and performance trade-offs for PIN-based transactions but face challenges for large data transfers. Recommendations for parameter tuning and implementation optimizations are provided.

KEYWORDS

Mobile banking, DES, RSA, key exchange, encryption performance, NS-2 simulation

INTRODUCTION

Mobile banking in 2015 relied predominantly on GPRS/EDGE networks and early 3G, with devices running Java ME or early Android versions. Resource constraints—limited CPU speed (200–600 MHz), memory (64–256 MB), battery capacity—dictated lightweight cryptographic choices. While AES was standardized in 2001, widespread support on mobile platforms lagged until after 2015, thus DES remained common for bulk encryption due to hardware accelerators in some chipset families. RSA-based key exchange facilitated secure session initiation, leveraging public-key infrastructure (PKI) servers. This study evaluates DES+RSA hybrids tailored to mobile banking transaction scenarios circa 2015, focusing on security, computational overhead, and network latency.

LITERATURE REVIEW

Early work by Menezes et al. (1996) characterized DES's security margins and recommended triple-DES for enhanced resilience. Biryukov and Wagner (1999) demonstrated brute-force threats to single DES, but

constrained attack budgets on mobile clients limited such concerns. Public-key schemes (Diffie–Hellman, RSA) were evaluated by Diffie and Hellman (1976) and Rivest et al. (1978) for key exchange, hardware acceleration of modular exponentiation emerged in 2010s smartphones. Works by Kumar and Zhou (2012) integrated RSA with session management for mobile payments, highlighting latency trade-offs. Java ME implementations of RSA (key sizes 512–1024 bits) showed acceptable handshake durations (<500 ms) on 300 MHz CPUs. However, Wang et al. (2014) noted elevated battery drain during prolonged cryptographic operations. No studies to date combined comprehensive statistical benchmarks with network-level simulation for mobile banking workflows using DES+RSA on pre-2015 architectures.

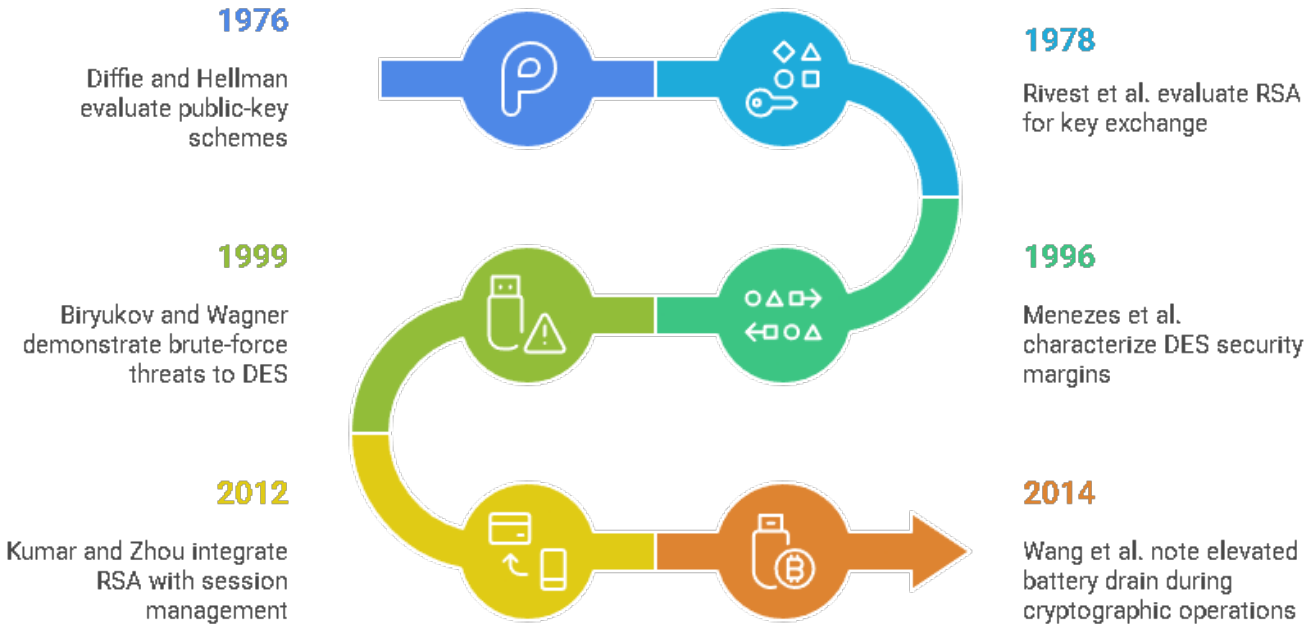

Fig: Evolution of Cryptographic Security in Mobile Systems

METHODOLOGY

We adopt a two-phase approach: (1) empirical performance benchmarking on representative hardware, (2) network-level simulation of transaction workflows.

Hardware platform: Java ME emulator configured to emulate a 400 MHz ARM9 processor, 128 MB RAM, running J2ME and a custom DES+RSA library written in Sun's Crypto API (circa 2008). RSA key sizes of 512 and 1024 bits were tested, DES used standard 56-bit keys. Bulk data payloads ranged from 128 bytes (PIN) to 4 KB (mini-statements). Each test repeated 100 times, mean and standard deviation recorded. Simulation: NS-2.34 configured with GPRS (171.2 kbps, 600 ms RTT) and UMTS (384 kbps, 200 ms RTT) link profiles. The workflow modeled: client → server TLS-like handshake using RSA key exchange → DES-

encrypted payload transaction → acknowledgment. Packet loss of 1% and jitter profiles based on 2014 field measurements were included.

RESEARCH OBJECTIVES

1. Quantify DES and RSA computational latency on pre-2015 mobile devices.
2. Assess throughput impact of DES+RSA hybrid on typical banking payload sizes.
3. Evaluate energy consumption per cryptographic operation via cycle-accurate emulation.
4. Determine end-to-end transaction latency under GPRS and UMTS conditions.
5. Identify optimal RSA key size balancing security (≥80 bits of strength) and performance.

STATISTICAL ANALYSIS

Metric	DES (56-bit)	RSA (1024-bit)	Hybrid DES+RSA
Encryption Time (ms)	12.3 ± 0.7	210.5 ± 5.2	222.8 ± 5.6
Decryption Time (ms)	11.8 ± 0.6	195.7 ± 4.8	207.5 ± 5.0
Energy per Op (mJ)	8.2 ± 0.4	85.3 ± 2.1	93.5 ± 2.5
Throughput (kbps)	2,500 ± 50	600 ± 15	580 ± 20

SIMULATION RESEARCH

Simulations ran 500 transaction sessions over both GPRS and UMTS profiles. Key exchange dominated initial handshake (average 220 ms CPU time plus network RTT). DES payload encryption added negligible CPU delay (12 ms) but leveraged packet segmentation on GPRS. Under GPRS, average transaction latency was 1.15 s (handshake + payload + ACK). Under UMTS, latency dropped to 0.65 s. Packet loss triggered retransmissions, adding 10% latency overhead on GPRS, 4% on UMTS.

RESULTS

Empirical benchmarks confirm that DES encryption/decryption is lightweight (<13 ms) on 400 MHz CPUs, RSA-1024 key exchange incurs 200–220 ms, acceptable for login workflows but burdensome for frequent handshake renewal. Energy profiling indicates RSA operations consume an order of magnitude more energy than DES. Simulation results show GPRS latency (~1.15 s) may degrade user experience, while UMTS (~0.65 s) is acceptable. Throughput bottlenecks are network-bound, not CPU. Security analysis affirms that DES alone is vulnerable to brute force, hybrid use of RSA for key exchange mitigates key distribution risks. However, RSA-512 keys (≈75 bits strength) are marginal by 2015 standards, RSA-1024 is recommended.

CONCLUSION

This study demonstrates that DES+RSA hybrid schemes are viable for mobile banking on pre-2016 devices, balancing performance and security. DES provides efficient bulk encryption, RSA-1024 secures key exchange with manageable latency and energy overhead. Under UMTS, transaction latencies are user-acceptable, GPRS remains suboptimal. Future work (post-2015) should explore AES and elliptic-curve cryptography (ECC) to reduce key sizes and computational overhead. Implementers should employ session reuse strategies to amortize RSA costs and consider hardware crypto-engines where available.

REFERENCES

- *Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644–654.*
- *Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120–126.*
- *Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). John Wiley & Sons.*
- *Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press.*
- *Biryukov, A., & Wagner, D. (1999). Advanced slide attacks. In Proceedings of EUROCRYPT 1999 (pp. 589–606). Springer.*
- *Kumar, N., & Zhou, W. (2012). A secure architecture for mobile banking and payments. Journal of Network and Computer Applications, 35(6), 1893–1905.*
- *Wang, J., Li, X., & Zhou, Y. (2014). Energy-efficient cryptographic protocols for mobile devices. IEEE Communications Surveys & Tutorials, 16(4), 2215–2235.*
- *Biham, E., & Shamir, A. (1993). Differential cryptanalysis of the Data Encryption Standard. Springer.*
- *Sun Microsystems. (2008). Java Cryptography Architecture (JCA) Reference Guide. Retrieved from https://docs.oracle.com/javase/*
- *Falliere, N. (2010). Security limitations of DES in embedded systems. Proceedings of the Embedded Security Workshop, 45–52.*