INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATION & MANAGEMENT

A Monthly Double-Blind Peer Reviewed (Refereed/Juried) Open Access International e-Journal - Included in the International Serial Directories

Indexed & Listed at:

Ulrich's Periodicals Directory ©, ProQuest, U.S.A., EBSCO Publishing, U.S.A., Cabell's Directories of Publishing Opportunities, U.S.A.,
Open J-Gage, India [link of the same is duly available at Inflibnet of University Grants Commission (U.G.C.)],
Index Copernicus Publishers Panel, Poland with IC Value of 5.09 & number of libraries all around the world.
Circulated all over the world & Google has verified that scholars of more than 2980 Cities in 165 countries/territories are visiting our journal on regular basis.
Ground Floor, Building No. 1041-C-1, Devi Bhawan Bazar, JAGADHRI – 135 003, Yamunanagar, Haryana, INDIA

http://ijrcm.org.in/

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the areas of Computer Science & Applications; Commerce; Business; Finance; Marketing; Human Resource Management; General Management; Banking; Economics; Tourism Administration & Management; Education; Law; Library & Information Science; Defence & Strategic Studies; Electronic Science; Corporate Governance; Industrial Relations; and emerging paradigms in allied subjects like Accounting; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Rural Economics; Co-operation; Demography: Development Planning; Development Studies; Applied Economics; Development Economics; Business Economics; Monetary Policy; Public Policy Economics; Real Estate; Regional Economics; Political Science; Continuing Education; Labour Welfare; Philosophy; Psychology; Sociology; Tax Accounting; Advertising & Promotion Management; Management Information Systems (MIS); Business Law; Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labour Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; International Relations; Human Rights & Duties; Public Administration; Population Studies; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism & Hospitality; Transportation Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic; Web Design and emerging paradigms in allied subjects.

Anybody can submit the **soft copy** of unpublished novel; original; empirical and high quality **research work/manuscript** ***anytime*** in ***M.S. Word format*** after preparing the same as per our **GUIDELINES FOR SUBMISSION**; at our email address i.e. infoijrcm@gmail.com or online by clicking the link **online submission** as given on our website (***FOR ONLINE SUBMISSION, CLICK HERE***).

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____________

THE EDITOR
IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF .**

(e.g. Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/Engineering/Mathematics/other, please specify)

DEAR SIR/MADAM

Please find my submission of manuscript entitled '____' for possible publication in your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published elsewhere in any language fully or partly, nor is it under review for publication elsewhere.

I affirm that all the author (s) have seen and agreed to the submitted version of the manuscript and their inclusion of name (s) as co-author (s).

Also, if my/our manuscript is accepted, I/We agree to comply with the formalities as given on the website of the journal & you are free to publish our contribution in any of your journals.

NAME OF CORRESPONDING AUTHOR:
Designation:
Affiliation with full address, contact numbers & Pin Code:
Residential address with Pin Code:
Mobile Number (s):
Landline Number (s):
E-mail Address:
Alternate E-mail Address:

NOTES:
a) The whole manuscript is required to be in ***ONE MS WORD FILE*** only (pdf. version is liable to be rejected without any consideration), which will start from the covering letter, inside the manuscript.
b) The sender is required to mentionthe following in the **SUBJECT COLUMN** of the mail:
New Manuscript for Review in the area of (Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/ Engineering/Mathematics/other, please specify)
c) There is no need to give any text in the body of mail, except the cases where the author wishes to give any specific message w.r.t. to the manuscript.
d) The total size of the file containing the manuscript is required to be below **500 KB**.
e) Abstract alone will not be considered for review, and the author is required to submit the complete manuscript in the first instance.
f) The journal gives acknowledgement w.r.t. the receipt of every email and in case of non-receipt of acknowledgment from the journal, w.r.t. the submission of manuscript, within two days of submission, the corresponding author is required to demand for the same by sending separate mail to the journal.

2. **MANUSCRIPT TITLE**: The title of the paper should be in a 12 point Calibri Font. It should be bold typed, centered and fully capitalised.

3. **AUTHOR NAME (S) & AFFILIATIONS**: The author (s) **full name**, **designation**, **affiliation** (s), **address**, **mobile/landline numbers**, and **email/alternate email address** should be in italic & 11-point Calibri Font. It must be centered underneath the title.

4. **ABSTRACT**: Abstract should be in fully italicized text, not exceeding 250 words. The abstract must be informative and explain the background, aims, methods, results & conclusion in a single para. Abbreviations must be mentioned in full.

5. **KEYWORDS**: Abstract must be followed by a list of keywords, subject to the maximum of five. These should be arranged in alphabetic order separated by commas and full stops at the end.

6. **MANUSCRIPT**: Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. It must be prepared on a single space and single column with 1" margin set for top, bottom, left and right. It should be typed in 8 point Calibri Font with page numbers at the bottom and centre of every page. It should be free from grammatical, spelling and punctuation errors and must be thoroughly edited.

7. **HEADINGS**: All the headings should be in a 10 point Calibri Font. These must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.

8. **SUB-HEADINGS**: All the sub-headings should be in a 8 point Calibri Font. These must be bold-faced, aligned left and fully capitalised.

9. **MAIN TEXT**: The main text should follow the following sequence:

   INTRODUCTION

   REVIEW OF LITERATURE

   NEED/IMPORTANCE OF THE STUDY

   STATEMENT OF THE PROBLEM

   OBJECTIVES

   HYPOTHESES

   RESEARCH METHODOLOGY

   RESULTS & DISCUSSION

   FINDINGS

   RECOMMENDATIONS/SUGGESTIONS

   CONCLUSIONS

   SCOPE FOR FURTHER RESEARCH

   ACKNOWLEDGMENTS

   REFERENCES

   APPENDIX/ANNEXURE

   It should be in a 8 point Calibri Font, single spaced and justified. The manuscript should preferably not exceed ***5000 WORDS***.

10. **FIGURES &TABLES**: These should be simple, crystal clear, centered, separately numbered &self explained, and **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. It should be ensured that the tables/figures are referred to from the main text.

11. **EQUATIONS**:These should be consecutively numbered in parentheses, horizontally centered with equation number placed at the right.

12. **REFERENCES**: The list of all references should be alphabetically arranged. The author (s) should mention only the actually utilised references in the preparation of manuscript and they are supposed to follow **Harvard Style of Referencing**. The author (s) are supposed to follow the references as per the following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parentheses.
- The location of endnotes within the text should be indicated by superscript numbers.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio" Ohio State University, Nigeria.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg, Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–22 June.

UNPUBLISHED DISSERTATIONS AND THESES

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.
- **ONLINE RESOURCES**
- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITES

- Garg, Bhavet (2011): Towards a New Natural Gas Policy, Political Weekly, Viewed on January 01, 2012 http://epw.in/user/viewabstract.jsp

ELLIPTIC CURVE CRYPTOGRAPHY

SANJEEV
RESEARCH SCHOLAR
DEPARTMENT OF COMPUTER SCIENCE
BHAGWANT UNIVERSITY
AJMER

DR. NAVEEN VERMA
ASST. PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE
GOVERNMENT COLLEGE
KAITHAL

ABSTRACT

We present the first known implementation of elliptic curve cryptography over F2p for sensor networks based on the 8-bit, 7.3828-MHz MICA2 mote. Through instrumentation of UC Berkeley's TinySec module, we argue that, although secret-key cryptography has been tractable in this domain for some time,there has remained a need for an efficient, secure mechanism for distribution of secret keys among nodes. Although public-key infrastructure has been thought impractical, we argue, through analysis of our own implementation for TinyOS of multiplication of points on elliptic curves, that public-key infrastructure is, in fact, viable for TinySec keys' distribution, even on the MICA2. We demonstrate that public keys can be generated within 34 seconds,and that shared secrets can be distributed among nodes in a sensor network within the same, using just over 1 kilobyte of SRAM and 34 kilobytes of ROM.

KEYWORDS

Cryptography, sensor networks.

INTRODUCTION

Wireless sensor networks have been proposed for such applications as habitat monitoring [1], structural health monitoring [2], emergency medical care [3], and vehicular tracking [4],all of which demand some combination of authentication,integrity, privacy, and security. Unfortunately, the state of the art has offered weak, if any, guarantees of these needs.The limited resources boasted by today's sensor networks appear to render them ill-suited for the most straightforward implementations of security protocols. Consider the MICA2 mote [5], designed by researchers at the University of California at Berkeley and fabricated by Crossbow Technology,Inc. This device offers an 8-bit, 7.3828-MHz ATmega 128L processor, 4 kilobytes (KB) of primary memory (SRAM), and 128 KB of program space (ROM). Such a device, given these resources, is seemingly unfit for computationally expensive or energy-intensive operations. For this reason has publickey cryptography often been ruled out for sensor networks as an infrastructure for authentication, integrity, privacy, and security [6]–[9], even despite its allowance for secure rekeying of mobile devices.

But such conclusions have been backed too infrequently by actual data. In fact, to our knowledge, little empirical research has been published on the viability of public-key infrastructure (PKI) for the MICA2, save for a cursory analysis of an implementation of RSA [10] and a recent comparison of RSA and elliptic curve cryptography (ECC) over F*p* [11].Our work aspires to fill this void. Through instrumentation of TinyOS, we first demonstrate that secret-key cryptography is tractable on the MICA2. By way of our own implementation of multiplication of points on elliptic curves, we then argue that PKI for secret keys' distribution is, in fact, tractable as well. Public keys can be generated within 34 seconds (sec),and shared secrets can be distributed within the same, using just over 1 KB of SRAM and 34 KB of ROM.We begin these arguments in Section II with an analysis of TinySec [6], TinyOS's existing secret-key infrastructure for the MICA2 based on SKIPJACK [12]. In Section III, we address shortcomings in that infrastructure with a look at an implementation of Diffie-Hellman for the MICA2 based on the Discrete Logarithm Problem (DLP) and expose weaknesses in its design for sensor networks.

BACKGROUND

Over the past 30 years, public key cryptography has become a mainstay for secure communications over the Internet and throughout many other forms of communications. It provides the foundation for both key management and digital signatures. In key management, public key cryptography is used to distribute the secret keys used in other cryptographic algorithms (e.g. DES). For digital signatures, public key cryptography is used to authenticate the origin of data and protect the integrity of that data. For the past 20 years, Internet communications have been secured by the first generation of public key cryptographic algorithms developed in the mid-1970's. Notably, they form the basis for key management and authentication for IP encryption (IKE/IPSEC), web traffic (SSL/TLS) and secure electronic mail.

This paper will outline a case for moving to elliptic curves as a foundation for future Internet security. This case will be based on both the relative security offered by elliptic curves and first generation public key systems and the relative performance of these algorithms. The two noteworthy first generation public key algorithms used to secure the Internet today are known as RSA and Diffie-Hellman (DH). The security of the first is based on the difficulty of factoring the product of two large primes. The second is related to a problem known as the discrete logarithm problem for finite groups. Both are based on the use of elementary number theory. Interestingly, the security of the two schemes, though formulated differently, is closely related.

SKIPJACK AND THE MICA2

TinyOS currently offers the MICA2 access control, authentication,integrity, and confidentiality through TinySec, a linklayer security mechanism based on SKIPJACK in cipher-block chaining mode. An 80-bit symmetric cipher, SKIPJACK is the formerly classified algorithm behind the Clipper chip, approved by the National Institute for Standards and Technology (NIST) in 1994 for the Escrowed Encryption Standard [13]. TinySec supports message authentication and integrity with message authentication codes, confidentiality with encryption, and access control with shared, group keys.The mechanism allows for an 80-bit key space, the benefit of which is that known attacks require as many 279 operations on average (assuming SKIPJACK isn't reduced from 32 rounds [14]).1 Moreover, as packets under TinySec include a 4-byte message authentication code (MAC), the probability of blind forgery is only 2−32. This security comes at a cost of just five bytes (B): whereas transmission of some 29-byte plaintext and its cyclic redundancy check (CRC) requires a packet of 36 B, transmission of that plaintext's ciphertext and MAC under TinySec requires a packet of only 41 B, as the mechanism borrows TinyOS's fields for Group ID (TinyOS's weak, default mechanism for access control) and CRC for its MAC.

Performance. The impact of TinySec on the MICA2's performance is reasonable. On first glance, it would appear that TinySec adds under 2 milliseconds (ms) to a packet's transmission time (Table I) and under 5 ms to a packet's roundtrip time to and from some neighbor (Table II). However,the apparent overhead of TinySec, 1,244 microseconds (μsec)on average, as suggested by transmission times, is nearly subsumed by the data's root mean square (1,094 μsec). Roundtrip times exhibit less variance, but tighter benchmarks are in order for TinySec's accurate analysis. Table III, then, offers results with yet less variance from finer instrumentation of TinySec: encryption of a 29-byte, random payload requires 2,190 µsec on average, and computation of that payload's MAC requires 3,049 µsec on average; overall,TinySec adds 5,239 } 18 µsec to a packet's computational requirements. It appears, then, that some of those cycles can be subsumed by delays in scheduling and medium access, at least for applications not already operating at full duty.The results of an analysis of the MICA2's throughput, without and with TinySec enabled, puts the mechanism's computational overhead for such applications into perspective: on average, TinySec may lower throughput of acknowledged packets by only 0.28 packets per second. These results appear in line with UC Berkeley's own evaluation of TinySec [15].

Memory. Of course, TinySec's encryption and authentication does come at an additional cost in memory. Per Table IV,TinySec adds 454 B to an application's .bss segment, 276 B to an application's .data segment, 7,076 B to an application's .text segment, and 92 B to an application's maximal stack size during execution. For applications that don't require the entirety of the MICA2's 128 KB of program memory and 4 KB of primary memory, then, TinySec is a viable addition.

Security. As with any cipher based only on shared secrets,TinySec is, of course, vulnerable to various attacks. After all,the MICA2 is intended for deployment in sensor networks. For reasons of cost and logistics, long-term, physical security of the devices is unlikely. Compromise of the network, therefore,reduces to compromise of any one node, unless, for instance,rekeying is possible. Pairwise keys among *n* nodes would cer-tainly provide some defense against compromises of individual nodes. But *n*2 80-bit keys would more than exhaust a node's SRAM for *n* as small as 20. A more sparing use of secret keys is in order, but secure, dynamic establishment of those keys, particularly for networks in which the positions of sensors may be transient, requires a chain or infrastructure of trust. In fact, the very design of TinySec requires as much for rekeying as well. Though TinySec's 4-byte initialization vector (IV) allows for secure transmission of some message as many as 232 times, that bound may be insufficient for embedded networks whose lifespans demand longer lasting security.2 Needless to say, TinySec's reliance on a single secret key prohibits the mechanism from securely rekeying itself.Fortunately, these problems of secret keys' distribution are redressed by public-key infrastructure. The sections that follow thus explore options for that infrastructure's design and implementation on the MICA2.

TABLE I: TRANSMISSION TIMES REQUIRED TO TRANSMIT A 29-BYTE, RANDOM PAYLOAD, AVERAGED OVER 1,000 TRIALS, WITH AND WITHOUT TINYSEC ENABLED. TRANSMISSION TIME IS DEFINED HERE AS THE TIME ELAPSED BETWEEN ENDMSG.SEND(・, ・, ・) AND SENDMSG.SENDDONE(). THE IMPLIED OVERHEAD OF TINYSEC ON TRANSMISSION TIME IS GIVEN AS THE DIFFERENCE OF THE DATA'S MEANS. THE ROOT MEAN SQUARE IS DEFINED AS

TABLE-1

	without TinySec	with TinySec
Median	72,904 µsec	74,367 µsec
Mean	74,844 µsec	76,088 µsec
Standard Deviation	24,248 µsec	24,645 µsec
Standard Error	767 µsec	779 µsec

Implied Overhead of TinySec	1,244 µsec
Root Mean Square	1,094 µsec

TABLE II: ROUND-TRIP TIMES REQUIRED TO TRANSMIT A 29-BYTE, RANDOM PAYLOAD, WITH AND WITHOUT TINYSEC ENABLED, FROM ONE NODE TO A NEIGHBOR AND BACK AGAIN, AVERAGED OVER 1,000 TRIALS. MORE PRECISELY, ROUND-TRIP TIME IS DEFINED HERE AS THE TIME ELAPSED BETWEEN SENDMSG.SEND(・, ・, ・) AND RECEIVEMSG.RECEIVE(・). THE IMPLIED OVERHEAD OF TINYSEC ON ROUND-TRIP TIME IS GIVEN AS THE DIFFERENCE OF THE DATA'S MEANS.

TABLE-2

	without TinySec	with TinySec
Median	145,059 µsec	149,290 µsec
Mean	147,044 µsec	152,015 µsec
Standard Deviation	30,736 µsec	31,466 µsec
Standard Error	972 µsec	995 µsec

Implied Overhead of TinySec	5,239 µsec
Root Mean Square	9 µsec

TABLE III: TIMES REQUIRED TO TO ENCRYPT A 29-BYTE, RANDOM PAYLOAD, AND TO COMPUTE THAT PAYLOAD'S MAC, AVERAGED OVER 1,000 TRIALS. THE IMPLIED OVERHEAD OF TINYSEC IS GIVEN DIFFERENCE OF THE _ DATA'S MEANS.

TABLE-3

Implied Overhead of TinySec	4,971 µsec
Root Mean Square	1,391 µsec

	encrypt()	computeMAC()
Median	2,189 µsec	3,038 µsec
Mean	2,190 µsec	3,049 µsec
Standard Deviation	3 µsec	281 µsec
Standard Error	0 µsec	9 µsec

TABLE IV: MEMORY OVERHEAD OF TINYSEC, DETERMINED THROUGH INSTRUMENTATION OF CNTTORFM, AN APPLICATION WHICH SIMPLY BROADCASTS A COUNTER'S VALUES OVER THE MICA2'S RADIO. THE .BSS AND .DATA SEGMENTS CONSUME SRAM WHILE THE .TEXT SEGMENT CONSUMES ROM. STACK IS DEFINED HERE AS THE MAXIMUM OF THE APPLICATION'S STACK SIZE DURING EXECUTION.

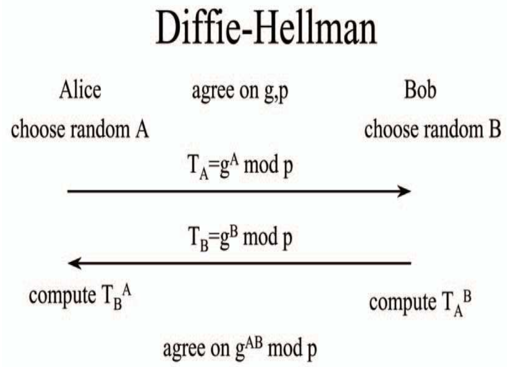
TABLE-4

	without TinySec	with TinySec	Difference
.bss	384 B	838	454 B
.data	4 B	280 B	276 B
.text	9,220 B	16,296 B	7,076 B
stack	105 B	197 B	92

DLP AND THE MICA2

With the utility of SKIPJACK-based TinySec thus motivated and the mechanism's costs exposed, we next examine DLP, on which Diffie-Hellman [16] is based, as an answer to the MICA2's problems of secret keys' distribution. DLP typically involves recovery of $x \in Zp$, given *p*, *g*, and *gx* (mod *p*),where *p* is a prime integer, and *g* is a generator of *Zp*. By leveraging the presumed difficultly of DLP, Diffie-Hellman allows two parties to agree, without prior arrangement, upon a shared secret, even in the midst of eavesdroppers, with perfectforward secrecy, as depicted in Fig. 2. Authenticated exchanges are possible with the station-to-station protocol (STS) [17], a variant of Diffie-Hellman.With a form of Diffie-Hellman, then, could two nodes thus establish a shared secret for use as TinySec's key. At issue, though, is the cost of such establishment on the MICA2. Inasmuch as the goal at hand is distribution of 80-bit TinySec keys, any mechanism of exchange should provide at least as much security. According to NIST [18], then, the MICA2's implementation of Diffie-Hellman should employ a modulus, *p*, of at least 1,024 bits and an exponent (*i.e.*, private key), *x*, of at least 160 bits (Table V).Unfortunately, on an 8-bit architecture, computations with 160-bit and 1,024-bit values are not inexpensive. However, modular exponentiation is not intractable on the MICA2. Table VI details the operations' memory usage.Of course, these measurements assume operation at full duty cycle, the energy requirements of which may be unacceptable,as the MICA2's lifetime decreases to just a few days at maximal duty cycle. Table VII reveals the MICA2's energy consumption for modular exponentiation: computation of 2*x* (mod *p*) appears to require 1.185 J. Roughly speaking, a mote could devote its lifetime to 51,945 such computations.3 Of course, these numbers might be improved (with, *e.g.*,hand-optimization). Unfortunately, these computations require not only time but also memory. Mere storage of a public key requires as many bits as is the modulus in use. Accordingly, *n* 1,024-bit keys would more than exhaust a node's SRAM for *n* as small as 32. Although a node is unlikely to have—or, at least, need—so many neighbors or certificate authorities for whom it needs public keys, Diffie-Hellman's relatively large key sizes are unfortunate in the MICA2's resource-constrained environment. A key of this size would not even fit in a single TinyOS packet.

FIG. 1



Typical exchange of a shared secret under Diffie-Hellman based on DLP [21].

TABLE V: STRENGTH OF DIFFIE-HELLMAN BASED ON DLP FOR VARIOUS MODULI AND EXPONENTS. "AN ALGORITHM THAT HAS A '*Y* ' BIT KEY, BUT WHOSE STRENGTH IS EQUIVALENT TO AN '*X*' BIT KEY OF SUCH A SYMMETRIC ALGORITHM IS SAID TO PROVIDE '*X* BITS OF SECURITY' OR TO PROVIDE '*X*-BITS OF STRENGTH'. AN ALGORITHM THAT PROVIDES *X* BITS OF STRENGTH WOULD, ON AVERAGE, TAKE 2*X*−1*T* TO ATTACK, WHERE *T* IS THE AMOUNT OF TIME THAT IS REQUIRED TO PERFORM ONE ENCRYPTION OF A PLAINTEXT VALUE AND COMPARISON OF THE RESULT AGAINST THE CORRESPONDING CIPHERTEXT VALUE." [18]

TABLE-5

Bits of Security	Modulus	Exponent
80	1,024	160
112	2,048	224
128	3,072	256
192	7,680	384
256	15,360	512

TABLE VI: MEMORY OVERHEAD OF MODULAR EXPONENTIATION, DETERMINED THROUGH INSTRUMENTATION OF AN IMPLEMENTATION OF DIFFIE-HELLMAN BASED ON DLP ON THE MICA2 WHICH COMPUTES 2*x* (MOD *p*), WHERE *x* IS A 512-BIT INTEGER AND *p* IS PRIME. THE .BSS AND .DATA SEGMENTS CONSUME SRAM WHILE THE .TEXT SEGMENT CONSUMES ROM. STACK IS DEFINED HERE AS THE MAXIMUM OF THE APPLICATION'S STACK SIZE DURING EXECUTION.

TABLE-6

	768-Bit Modulus	1,024-Bit Modulus
.bss	852 B	980 B
.data	102 B	134 B
.text	11,334 B	11,350 B
stack	136 B	136 B

TABLE VII: ENERGY CONSUMPTION OF MODULAR EXPONENTIATION, DETERMINED THROUGH INSTRUMENTATION OF AN IMPLEMENTATION OF DIFFIE-HELLMAN BASED ON DLP ON THE MICA2 WHICH COMPUTES 2*x* (MOD *p*), WHERE *x* IS A 160-BIT INTEGER AND *p* IS A 1,024-BIT PRIME.

TABLE-7

	1,024-Bit Modulus, 160-Bit Exponent
Total Time	54.1144 sec
Total CPU Utilization	3.9897 × 108 cycles
Total Energy	1.185 Joules

ELLIPTIC CURVE SECURITY AND EFFICIENCY

The majority of public key systems in use today use 1024-bit parameters for RSA and Diffie-Hellman. The US National Institute for Standards and Technology has recommended that these 1024-bit systems are sufficient for use until 2010. After that, NIST recommends that they be upgraded to something providing more security. The question is what should these systems be changed to? One option is to simply increase the public key parameter size to a level appropriate for another decade of use. Another option is to take advantage of the past 30 years of public key research and analysis and move from first generation public key algorithms and on to elliptic curves.

One way judgments are made about the correct key size for a public key system is to look at the strength of the conventional (symmetric) encryption algorithms that the public key algorithm will be used to key or authenticate. Examples of these conventional algorithms are the Data Encryption Standard (DES) created in 1975 and the Advanced Encryption Standard (AES) now a new standard. The length of a key, in bits, for a conventional encryption algorithm is a common measure of security. To attack an algorithm with a k-bit key it will generally require roughly 2k-1 operations. Hence, to secure a public key system one would generally want to use parameters that require at least 2k-1 operations to attack. The following table gives the key sizes recommended by the National Institute of Standards and Technology to protect keys used in conventional encryption algorithms like the (DES) and (AES) together with the key sizes for RSA, Diffie-Hellman and elliptic curves that are needed to provide equivalent security.

TABLE – 8: NIST RECOMMENDED KEY SIZES

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

To use RSA or Diffie-Hellman to protect 128-bit AES keys one should use 3072-bit parameters: three times the size in use throughout the Internet today. The equivalent key size for elliptic curves is only 256 bits. One can see that as symmetric key sizes increase the required key sizes for RSA and Diffie-Hellman increase at a much faster rate than the required key sizes for elliptic curve cryptosystems. Hence, elliptic curve systems offer more security per bit increase in key size than either RSA or Diffie-Hellman public key systems.

Security is not the only attractive feature of elliptic curve cryptography. Elliptic curve cryptosystems also are more computationally efficient than the first generation public key systems, RSA and Diffie-Hellman. Although elliptic curve arithmetic is slightly more complex per bit than either RSA or DH arithmetic, the added strength per bit more than makes up for any extra compute time. The following table shows the ratio of DH computation versus EC computation for each of the key sizes listed in Table 8.

TABLE 9: RELATIVE COMPUTATION COSTS OF DIFFIE-HELLMAN AND ELLIPTIC CURVES[1]

Security Level (bits)	Ratio of DH Cost : EC Cost
80	3:1
112	6:1
128	10:1
192	32:1
256	64:1

Closely related to the key size of different public key systems is the channel overhead required to perform key exchanges and digital signatures on a communications link. The key sizes for public key in Table 8 (above) is also roughly the number of bits that need to be transmitted each way over a communications channel for a key exchange[2]. In channel-constrained environments, elliptic curves offer a much better solution than first generation public key systems like Diffie-Hellman.

In choosing an elliptic curve as the foundation of a public key system there are a variety of different choices. The National Institute of Standards and Technology (NIST) has standardized on a list of 15 elliptic curves of varying sizes. Ten of these curves are for what are known as binary fields and 5 are for prime fields. Those curves listed provide cryptography equivalent to symmetric encryption algorithms (e.g. AES, DES or SKIPJACK) with keys of length 80, 112, 128, 192, and 256 bits and beyond as shown in table-9.

For protecting both classified and unclassified National Security information, the National Security Agency has decided to move to elliptic curve based public key cryptography. Where appropriate, NSA plans to use the elliptic curves over finite fields with large prime moduli (256, 384, and 521 bits) published by NIST.

The United States, the UK, Canada and certain other NATO nations have all adopted some form of elliptic curve cryptography for future systems to protect classified information throughout and between their governments. The Cryptographic Modernization Initiative in the US Department of Defense aims at replacing almost 1.3 million existing equipments over the next 10 years. In addition, the Department's Global Information Grid will require a vast expansion of the number of security devices in use throughout the US Military. This will necessitate change and rollover of equipment with all major US allies. Most of these needs will be satisfied with a new generation of cryptographic equipment that uses elliptic curve cryptography for key management and digital signatures.

CONCLUSION

Elliptic Curve Cryptography provides greater security and more efficient performance than the first generation public key techniques (RSA and Diffie-Hellman) now in use. As vendors look to upgrade their systems they should seriously consider the elliptic curve alternative for the computational and bandwidth advantages they offer at comparable security. Despite claims to the contrary, public-key infrastructure appears viable on the MICA2, certainly for infrequent distribution of shared secrets. Although our implementation of ECC in 4 KB of primary memory on this 8-bit, 7.3828-MHz device offers room for further optimization, even a minute's worth of computation every 232 transmissions (or every day or every week) seems reasonable for re-keying.The need for PKI's success on the MICA2 seems clear.TinySec's shared secrets do allow for efficient, secure communications among nodes. But such devices as those in sensor networks, for which physical security is unlikely, require some mechanism for secret keys' distribution.In that it offers equivalent security at lower cost to memory and bandwidth than does Diffie-Hellman based on DLP, a public-key infrastructure for key distribution based on elliptic curves is an apt, and viable, choice for TinyOS on the MICA2.

REFERENCES

1. A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application Driver for Wireless Communications Technology," 2001.
2. V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, E. Carryer, T. W. Kenny, K. H. Law, and Y. Lei, "Two-tiered wireless sensor network architecture for structural health monitoring," SPIE's 10th Annual International Symposium on Smart Structures and Materials, March 2009.
3. Vital Dust: Wireless Sensor Networks for Emergency Medical Care, http://www.eecs.harvard.edu/~mdw/proj/vitaldust/.
4. NEST Challenge Architecture, August 2002.
5. I. Crossbow Technology, "MICA2: Wireless Measurement System," http://www.xbow.com/Products/Product pdf files/Wireless pdf/6020-0042-0%4 A MICA2.pdf.
6. C. Karlof, N. Sastry, and D. Wagner, "TinySec: Link Layer Security for Tiny Devices," http://www.cs.berkeley.edu/~nks/tinysec/.
7. A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," Communications of the ACM, vol. 47, no. 6, pp. 53–57, June 2011.
8. A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, "SPINS:Security Protocols for Sensor Networks," in Mobile Computing and Networking, 2001, pp. 189–199.
9. C. Karlof, N. Sastry, and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks." Baltimore, Maryland:Second ACM Conference on Embedded Networked Sensor Systems,November 2012.
10. R. Watro, "Lightweight Security for Wireless Networks of Embedded Systems," http://www.is.bbn.com/projects/lws-nest/bbn nest apr 03.ppt, May 2003.
11. N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs." Boston,Massachusetts: 6th International Workshop on Cryptographic Hardware and Embedded Systems, August 2011.
12. Computer Security Division, SKIPJACK and KEA Algorithm Specifications,National Institute of Standards and Technology, May 1988.
13. National Institute of Standards and Technology, "Federal Information Processing Standards Publication 185: Escrowed Encryption Standard (EES)," February 2010. [Online]. Available: {http://www.itl.nist.gov/fipspubs/fip185.htm}
14. E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials," *Lecture Notes in Computer Science*, vol. 1592, pp. 12–23, 1999. [Online]. Available:citeseer.nj.nec.com/biham99cryptanalysis.html
15. Naveen Sastry, University of California at Berkeley, personal correspondence.
16. W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654,1976. [Online]. Available: citeseer.nj.nec.com/diffie76new.html
17. W. Diffie, P. C. van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes, and Cryptography*, vol. 2,no. 2, pp. 107–125, 2013.

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Computer Application & Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you tosupply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mail**infoijrcm@gmail.com** for further improvements in the interest of research.

If youhave any queries please feel free to contact us on our E-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-
Co-ordinator

DISCLAIMER

ABOUT THE JOURNAL

In this age of Commerce, Economics, Computer, I.T. & Management and cut throat competition, a group of intellectuals felt the need to have some platform, where young and budding managers and academicians could express their views and discuss the problems among their peers. This journal was conceived with this noble intention in view. This journal has been introduced to give an opportunity for expressing refined and innovative ideas in this field. It is our humble endeavour to provide a springboard to the upcoming specialists and give a chance to know about the latest in the sphere of research and knowledge. We have taken a small step and we hope that with the active co-operation of like-minded scholars, we shall be able to serve the society with our humble efforts.

Our Other Journals