A Scheme for the Generation of Strong Cryptographic Key Pairs based on ICMetrics

Ruhma Tahir, Huosheng Hu, Dongbing Gu, Klaus McDonald-Maier
School of Computer Science and Electronic Engineering
University of Essex
Colchester, United Kingdom

Gareth Howells
School of Engineering and Digital Arts
University of Kent
Canterbury, United Kingdom

Abstract— **This paper presents a scheme for the generation of strong high entropy keys based on ICMetrics. ICMetrics generates the security attributes of the sensor node based on measurable hardware and software characteristics of the integrated circuit. This work is based on key derivation functions to derive cryptographic key pairs from ICMetrics values. The proposed ICMetrics based key derivation function makes use of ICMetrics basis numbers and authentication tokens from the trusted third party to generate high entropy public/private key pairs. The proposed approach makes use of key stretching using SHA-2 and performs multiple iterations of the proposed key derivation function to generate strong high entropy keys of sufficient length, so as to prevent exhaustive search attacks. The novelty of this work lies in the fact that the entire key generation scheme has been designed keeping in mind the construction principles of ICMetrics, which does not store keys but computes these for every session based on ICMetrics value, therefore use of a random value anywhere in the protocol will compromise the purpose of ICMetrics. The proposed scheme generates high entropy key pairs while concealing the original ICMetrics data, such that it is impossible to recover the ICMetrics basis data in the system.**

Keywords- ICMetrics(Integrated Circuit Metrics), Hermite Normal Form (HNF), Trusted Third Party(TTP), key stretching, key derivation function

I. INTRODUCTION

Embedded computing systems play a major part in various aspects of life, ranging from their use in single user applications to large scale processes [1]. Examples of their use in large scale processes could be sensor nodes aggregating sensed data and transmitting it to a centralized server or simply their use in handheld devices for a range of single user applications. The widespread deployment of high speed/bandwidth wireless networks has emerged many embedded system applications as well as ported many existing applications to embedded systems. However with the increase in data communications using these embedded systems, the requirement for applications security has become a major concern [2].

The security of all sensitive applications mainly depends on the proper generation and protection of its cryptographic keys, since the underlying encryption/decryption algorithms are published and globally known. Therefore there are inherent risks if proper keying materials and key management is not used for secure operations; this can result in compromise of encryption keys that will seriously affect the integrity, confidentiality, and availability of communications. Without proper generation and handling of keys, keys could be easily guessed, modified, or substituted by unauthorized personnel who could then intercept sensitive communications. Embedded systems have limitations in terms of power, memory and processing speed [3]; therefore care needs to be taken to choose cryptographic mechanisms for key generation, and to determine key sizes according to individual applications [4]. Cryptographic algorithms that are used to provide secure communication in embedded system applications depend on the use of stored encryption/decryption keys. These algorithms have the inherent disadvantage that the compromise of an embedded system device, can lead to key/ secret information being revealed to the adversaries, which can ultimately result in even the entire network being overtaken and important data being revealed [5].

Integrated Circuit Metrics or ICMetrics [8] is an alternative to stored encryption/decryption keys that uses unique measurable properties and features of a hardware device to generate a basis number. ICMetrics is very similar to biometrics where human properties and features are used to uniquely identify and associate an identity with a person [9]. However any number in its raw form cannot serve as a key for cryptographic operations; since the extracted keying material might be too short or have low entropy, which would make it easy to compromise for an attacker. Therefore the generated cryptographic keys must possess sufficient key entropy and length, before they can qualify to be kept as a key for secure cryptographic operations.

In this paper we address this issue and propose a key generation mechanism for ICMetrics basis number that generates high entropy public/private key pair with sufficient levels of security. The proposed key generation mechanism based on ICMetrics data has the following features that address the key management issues in embedded system environments:

1) To safeguard against issues related to key compromise, the proposed architecture is based on the use of ICMetrics

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council under grant EP/K004638/1 and the EU Interreg IV A 2 Mers Seas Zeeën Cross-border Cooperation Programme – SYSIASS project: Autonomous and Intelligent Healthcare System (project's website http://www.sysiass.eu/).

values for key pair generation [10]. ICMetrics generates keys based on the hardware/software characteristics and the specification of the device which provides an effective means to address the issues related to the secrecy of key. Our scheme binds a cryptographic key with the device's ICMetrics information.

2) For the purpose of authentication/ identification of the device, in our scheme the high entropy key pair is formed by combining the ICMetrics generated basis number with a partial key generated by the Trusted Third Party (TTP). Both of these secret values are combined through a SHA-2 based key derivation function [16].

3) As stated above, the main challenge with the ICMetrics generated basis number is the entropy and length of the generated secret value. So our proposed design generates a high entropy key of sufficient length using SHA-2 based key derivation function. We iterate through multiple rounds of the SHA-2 based hash function [24] to stretch the secret value to the required length, thereby also generating a key with high entropy.

4) Lastly, our scheme generates a public key corresponding to the generated high entropy private key by computing the Hermite Normal Form of the private key [6]. The Hermite Normal Form is particularly suitable for public key generation since its unique, non-reversible and doesn't require any random values/ functions for operations [7].

The remainder of this paper is organized as follows; in section 2 we discuss characteristics of a strong cryptographic key and how weak keys generated by ICMetrics feature values pose a threat to security of applications. Section 3 introduces the security primitives on which the design of our proposed architecture rests. Section 4 explains the network topology for our proposed design. The design of our proposed framework for the generation of strong high entropy keys is presented in section 5. Section 6 presents the analysis of its security. The conclusion and future work for our paper is presented in section 7.

II. Strong Cryptographic Keys and ICMetrics

A. Strong Cryptographic Keys

The security of an application mainly depends on the secure generation and usage of cryptographic keys, so that they are safeguarded against all sorts of attacks during their usage in secure operations. As stated above, the generated cryptographic keys must also have certain properties to qualify as a key to be used for secure operations, since any number in its raw form cannot serve as a key for cryptographic operations. A longer cryptographic key is more secure, since it makes it harder for the adversary to break the encrypted text [16][19]. Although strong cryptographic functions provide data privacy and protection but without secure keys the integrity and confidentiality of an application is still at risk.

A major threat in security applications having weak/ low entropy keys is the launch of a brute force/ exhaustive search attack [15]. A brute force attack is a very common methodology adopted by adversaries to break cryptosystems with strong cryptographic operations but weak keys; whereby instead of finding weaknesses in the encryption system, the attacker tries to crack the cryptographic key used for performing the cryptographic operations. From an attacker's perspective, longer keys are harder to break compared to shorter keys, since the resources required to launch a brute force attack grow exponentially with an increase in key size [19].

B. ICMetrics

Traditionally cryptographic algorithms have always relied on the use of stored keys for functioning of the network. However the use of stored keys to enable secure communication is threatened by the fact that, if the key used to encrypt the data is compromised, it will result in loss of data that is encrypted using the compromised key.

ICMetrics (Integrated Circuit metrics) makes use of system level characteristics to provide identification to the system [12]. It generates encryption key from measurable properties of a given hardware device such as hardware/ software characteristics and specification of the node; similar to the way biometrics extracts human features.

After each ICMetrics key generation stage the produced ICMetric key [13] is temporary and exists only locally, and the reproduction of the ICMetric key once again takes place from measurable characteristics of the integrated circuit [10]. The ICMetric key is generated each time it is requested for identification or encryption functions. It consists of measuring the system features, applying the normalisation maps and combining the various feature values to generate a basis number for identifying the system. The individual feature values are be combined using two possible feature combination techniques; each generating a different sized key of varying stability. The feature addition-combination technique generates a stable but small basis number by adding individual feature values, while the feature concatenation-combination technique generates a long but less stable basis number by concatenating individual feature values. The resultant basis number is used for the subsequent derivation of the key required for the actual encryption process. Modifications to either the software executing on a given device, or to its hardware, will cause variations of the feature measurable characteristics and therefore the derived basis value. This in turn will mean that the system has been tampered and will not be able to take part in future operations [5].

In terms of issues related to key compromise of a device, ICMetrics proves to be breakthrough technology, thereby generating keys for secure identification of the device. However the ICMetrics generated keys can be weak, since they maybe short and of low entropy; and could infact pose a severe threat to the security of the system. Since a stable ICMetrics generated basis number is small in size with low entropy, it makes the underlying device open to attacks such as brute force and exhaustive search attacks. These attacks can in turn completely spoil the security of the application and even lose the essence/advantage of ICMetrics. Therefore, our proposed scheme intends to resolve the issue of weak keys and generates

strong keys with high entropy and sufficient length, thereby making ICMetrics a very viable option for secure systems.

III. Security Primitives

All security applications require proper generation and maintenance of keys during their lifecycle. The security of an application also depends on the strength of the cryptographic key. In the following section we briefly describe the various security primitives that we have employed in our proposed scheme for the generation of strong key pairs based on ICMetrics:

A. Key Generation and Public Key Cryptography

Key generation is one of the most sensitive cryptographic functions. Cryptographic keys that form a part of symmetric key schemes have a single secret key that is shared between communicating parties and is used for all encryption/ decryption operations. Cryptographic keys that form a part of asymmetric key schemes have two parts; a private key and a corresponding public key. A public key can be publically distributed whereas a private key has to be kept secret. The public key is used by other parties to send messages securely to the person that generated the key pair or to verify digital signatures generated by the person who generated the key pair. The private key is used to decrypt messages or to generate digital signatures[26].

B. Key Derivation Function

Key Derivation Function (KDF) is a special transformation function that can bring a number in raw form to a form that can be securely used as a key for secure operations [16]. This transformation on the raw number will safeguard the key against brute force attacks and exhaustive search attacks. This special mechanism that transforms raw keys to secure cryptographic keys of a required length, and also generates keys having sufficient entropy and randomness is called a key derivation function [17]. This function is essential in all security applications and generates keys with high entropy that can be safely used in security critical applications. The key derivation function, takes the raw password and a random salt as input, and applies multiple iterations of a function H (such as hash or block cipher); to generate keys with high entropy.

C. Key Stretching

Key-stretching is a method to hinder an attacker's ability to reproduce a key derivation function [19]. Key stretching makes use of iterated hashing to generate keys of a particular length with high entropy. Key stretching strengthens the key against brute force attacks by increasing key length and entropy thereby making it unfeasible for the attacker to launch brute force attacks.

Although key derivation functions and key stretching give a sense of similarity, since they are both based on hash generation functions to generate high entropy keys [18]. However their design principles are significantly different, since key stretching tends to derive long keys by concatenating the output of each hash function iteration while Key Derivation Functions (KDF) iteratively hash the same input key to produce high entropy key.

D. SHA-2

A hash function [20] is a deterministic function that takes an arbitrary length bit string as input and outputs a fixed length bit string called a hash value. The length of generated hash value depends on the employed hash function [21]. The well-known hash algorithms include MD 5, SHA1, SHA2 etc.; each having variants of differing hash value lengths for an input block [22][23][24].

SHA-2 [24] is a set collision resistant cryptographic hash functions proposed by the National Security Agency. It is the second in Secure Hash Algorithm (SHA) series designed by the National Security Agency. SHA-2 is a stronger hash function from its predecessor SHA-1 [22]; and is the standard adopted in many security applications and protocols such as TLS, IPSec, etc. SHA-2 has four hash variants of varying sized digests, namely SHA-224, SHA-256, SHA-384 and SHA-512. SHA-224 outputs a 224 bits digest, computing on block sizes of 512 bits in each of the 64 rounds of the hash computation operation. SHA-256 outputs a 256 bits digest, working with 512 bit blocks in each of the 64 rounds of hash computation operation. SHA-384 and SHA-512 both work with block sizes of 1024 bits in each of the 80 rounds while giving digest sizes of 384 bits and 512 bits respectively [24].

E. Hermite Normal Form

In our research, the public key for the generated high entropy private key is generated based on Hermite Normal Form (HNF) of the high entropy private key. Every matrix has a unique corresponding Hermite Normal Form 'H' and can be bought into its corresponding Hermite Normal Form by a sequence of elementary column operations. A non-singular square matrix 'H' is said to be in Hermite Normal Form (HNF) if it has the following properties[6][7]:

1) $h_{ij} = 0$ for $i < j$ (i.e., H is lower triangular);
2) $0 \leq h_{ij} < h_{ii}$ for $i>j$ (i.e., H is non-negative and each row has a unique maximum entry, which is on the main diagonal)

IV. Network Topology

Our proposed scheme is an idea for networked environments and all devices forming part of the network have trust in a trusted third party (TTP), as shown in Fig. 1. A trusted third party enables entities that never had contact before to interact securely and confidentially. We discuss our key generation scheme in terms of communication within a single cluster, although our scheme can be easily extended to multiple clusters connected together through multiple TTP's. Authentication between the trusted third party and the entities takes place based on traditional authentication protocols [25-30] and is not the focus of this paper. The focus of this research is to propose a scheme for the generation of strong keys based on ICMetrics, so we haven't formally explained the process of authentication between the entities and the

trusted third party, and therefore we simply assume that all communications take place authentically and confidentially.

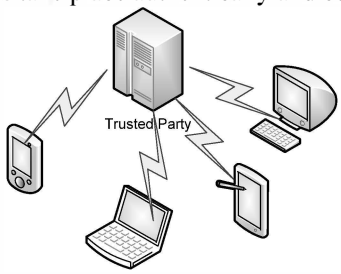

Figure 1. Network Topology for Proposed Architecture

V. THE PROPOSED SECURITY FRAMEWORK

In our design, we aim to improve the security of ICMetrics secret key by proposing a scheme that generates strong key pairs based on ICMetrics basis number. The proposed scheme is an idea for the generation of high entropy ICMetrics keys of sufficient length, which can be used for secure cryptographic operations in various applications. The following section details the six steps involved for the generation of strong public/private key pair based on ICMetrics extracted feature values:

A. Key Setup

The first phase of the proposed scheme requires each device that is part of the network to generate an ICMetrics basis number based on the extracted feature values. In [12], Papoutsis et al. propose two possibilities for combining the feature values of a device based on ICMetrics; concatenation of feature values or addition of feature values to generate a basis number, and hence serve as the secret key. The authors also show that a basis number generated as a result of concatenation of feature values although generates a long secret key but the secret key is not stable. On the other hand, while the basis number generated as a result of addition of feature values is short (35 bits) but is more stable. The ideal case would be to have a basis number that is stable as well as long, so that it qualifies for becoming a secret key during secure network operation. In our proposed scheme we take a stable but short basis number that was generated as a result of addition of feature values rather than concatenation, so each of the extracted feature values are added to generate a short but stable secret key/basis number.

Therefore, the key setup algorithm generates the master private key for the trusted third party (TTP) and all other entities that form a part of the network. The ICMetrics generated basis number of the trusted entity serves as its master private key.

Master Private Key of TTP 'MPr_t' = Basis Number using ICMetrics of Device

Similarly all entities with identities ID_1, ID_2, …, ID_n also generate their its master private key's MPr_1,MPr_2,…, MPr_n respectively based on their ICMetrics basis number and hence use their master private key for further operations as shown in figure as shown in Fig. 2.

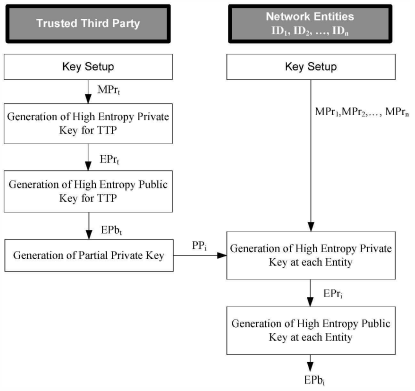

Figure 2. Functional Diagram of Proposed Design

B. Generation of High Entropy Private Key for TTP

This algorithm is responsible for generation a of high entropy private key for the TTP. The first step of this algorithm involves the TTP computing its master public key 'MPb_t' based on its master private key 'MPr_t' using the following

Master Public Key of TTP 'MPb_t'= Hermite-Normal-Form (Master Private Key of TTP 'MPr_t')

Then the TTP combines its master private key 'MPr_t' and HNF(MPr_t || MPb_t) using SHA-2 to generate high entropy private key 'EPr_t' for TTP that can be used for secure onward operation, as shown in Fig.3. Both 'MPr_t' and HNF (MPr_t || MPb_t) are combined via a SHA-2 based key stretching and derivation algorithm. The purpose of the SHA-2 based key stretching algorithm is to combine and thus stretch the key, so that it qualifies for use in secure operations. We propose the use of SHA-2 for the purpose of key stretching which will bring an increase in the key length as well as increase the entropy of the key.

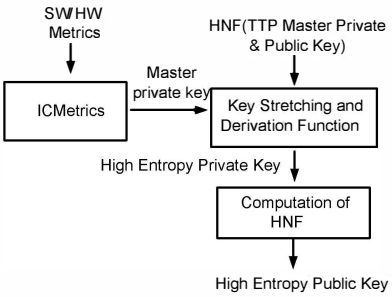

Figure 3. High Entropy Key Generation at TTP

The proposed SHA-2 based key stretching algorithm takes the ICMetrics generated basis number as input and after performing multiple iterations of each round of SHA-2; it produces a longer stretched high entropy key as shown in Table I. The value of number of iterations 'n' of SHA-2 has to be selected by setting a trade-off between security and efficiency; since greater number of iterations of the hash function will translate directly into an increase in entropy of the generated key. Therefore the value of n can be selected based on application requirements as shown below in table 1.

Table I. Proposed Key Derivation and Stretching Algo. for TTP Priv. Key

$X_0 = SHA\text{-}2\ (MPr_t, HNF\ (MPr_t \| MPb_t))$

For t = 1 to n

$X_i = SHA\text{-}2\ (MPr_i, X_{t-1})$

$X_0 = X_0 \| X_i$

High Entropy Private Key 'EPr_t' = X_0

Now the generated key 'X_0' is broken into required key length blocks; starting from the right and appending zeros to the left most block to make its size compatible with rest of the blocks. Then finally all the blocks are XORed to obtain the required sized key, which gives the final high entropy private key for the TTP.

C. *Generation of High Entropy Public Key for TTP*

This step involves the TTP computing its corresponding public key from the high entropy private key 'EPr_t' using:

Public Key of TTP 'EPb_t'= Hermite-Normal-Form(High Entropy Private Key of TTP 'MPr_t')

Finally the TTP has a high entropy public/private key pair generated that it can use for onward operations/ communication with other entities in the network.

D. *Generation of Partial Private Key*

This algorithm is run by the TTP in which it generates a partial key for each network entity based on entity's publically known identity information. We use the MAC address of each device as publically known identity information.

1) Compute Qi = SHA-2 (ID_i)
2) Output partial private key 'PP_i'=(EPr_t)x(Q_i)

The generated partial private key 'PPi' generated by the TTP is sent to entity ID_i. This process of supplying partial private keys takes place confidentially and authentically, the KGC ensures that the partial private keys are delivered securely to the correct entities.

E. *Generation of High Entropy Private Key at each Entity*

The user with identity ID_i, combines both the TTP generated partial private key 'PP_i' and its own master private key 'MPr_i' to generate high entropy private key 'EPr_i' for ID_i, for secure onward operations, as shown in Fig. 4.

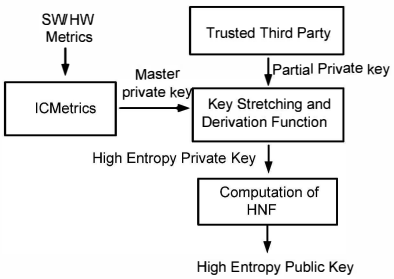

Figure 4. High Entropy Key Generation at each Participating Entity

Both PP_i and MPr_i are combined via SHA-2, as shown in Table II. The purpose of the SHA-2 based key stretching and derivation algorithm is to combine and thus stretch the key, so that it qualifies for use in secure operations. We propose the use of SHA-2 for the purpose of key stretching and derivation since they will both bring an increase in the key length as well as increase the entropy of the key. The SHA-2 based key stretching and derivation algorithm takes the ICMetrics generated basis number as input and after performing multiple iterations of SHA-2 rounds, it produces a longer stretched high entropy key.

Table II: Proposed Key Derivation and Stretching Algo. for Entity's Priv. Key

$X_0 = SHA\text{-}2(MPr_i, PP_i)$

For t = 1 to n

$X_i = SHA\text{-}2(MPr_i, X_{t-1})$

$X_0 = X_0 \| X_i$

High Entropy Private Key 'EPr_i' = X_0

Now the generated key 'X_0' is broken into blocks of equal size according to the required key length; starting from the right and appending zeros to the left most block to make its size compatible with rest of the blocks. Then finally all blocks are XORed to generate the required sized key, which gives the final high entropy private key for the entity ID_i. Exclusive Ors add an extra layer of protection but the actual security lies in the iterations of the hashing function.

F. *Generation of High Entropy Public Key at each Entity*

Then it computes public key from the high entropy private key 'EPr_i' using the following:

Public Key of ID_i 'EPb_i'= Hermite-Normal-Form(High Entropy Private Key of ID_i 'EPr_t')

Finally the entity ID_i now has a high entropy public/private key pair generated that it uses for secure onward operations/ communication with other entities in the network.

VI. SECURITY ANALYSIS

In the proposed scheme we have combined the security advantages of ICMetrics secret key and security properties of SHA-2 for the generation of a strong private key. The security of our scheme is based on the mixing of a partial key from the TTP with the ICMetrics basis number, and rehashing it a number of times based on SHA-2, to produce a strong public/ private key pair. This feature ensures authenticity/ identification of participating entities and also assures the origin of information; since only entities that have been assigned a partial key from TTP can communicate with other entities in the network. To regenerate the public/ private key pair, the generating entity must have the knowledge of both the user partial key and secret basis number. Knowing only one of them does not allow the generation of key pairs. This safeguards the network from attackers, since only

authenticated entities that have been issued a partial secret by the TTP can form part of network. The SHA-2 based key stretching and derivation operation on the ICMetrics secret key, results in a longer and high entropy private key that helps safeguards against brute force/ exhaustive search attack.

The second phase of our scheme, generates a corresponding strong public key by computing the HNF. The corresponding public key of a generated strong private key is generated by computing the HNF of the private key. The Hermite Normal Form of a number is unique and non-reversible, therefore it proves to be a very good choice for the computation of the corresponding public key. Each participating entity's public key is made available to other entities by transmitting it along with messages or by placing it in a public directory. But no further security is applied to the protection of A's public key. This idea helps preserve the security properties of ICMetrics generated keys and at the same time generates high entropy key pairs for use in future operations.

The proposed scheme also has its efficiency advantages, since the high entropy private key of an entity is computed every time a session starts and removed from memory once the session is over, whereas the public key is computed only once when an entity joins the network and stays with the participating nodes. This eliminates tremendous amounts of computation/ communication required for computing/ distributing the public keys whenever a new session starts.

VII. Conclusion and Future Work

A framework for generation of strong high entropy keys of sufficient length for ICMetrics secret key is introduced in this paper. The proposed scheme effectively combines the functionalities of ICMetric keys coupled with authentication tokens from the trusted third party, thereby providing authentication and identification of the device. In this paper, we have designed a password-based key derivation function, in which we employ SHA-2 for stretching the ICMetrics key using partial key from the TTP, to generate a high entropy ICMetrics private key. The high entropy public key corresponding to the high entropy key is computed by calculating the Hermite Normal Form of the high entropy private key. The proposed scheme has been very carefully tuned with the underlying requirements of ICMetrics. By employing our key generation scheme we are able to prevent most of the depicted threats while considering the constraints that embedded systems demands. The design decisions of our architecture have a number of important consequences and make it particularly suitable for resource constrained environments, which require elimination of additional communication and resource intensive operations while providing the required security.

Our future plan is to evaluate the scheme proposed here through experiments and analysis, thus benchmarking the results against existing key generation schemes in resource constrained environments. We are confident that our proposed architecture will be an efficient and viable solution for secure generation of strong high entropy key pairs. We also plan to design a protocol for secure communication based on the generated high entropy key pairs.

References

[1] I. Ryu, "Issues and Challenges in Developing Embedded Software for Information Appliances and Telecommunication Terminals", Proceedings of the ACM SIGPLAN 1999, LCTES'99, Vol. 34, Issue7, July 1999, pp. 104-120.

[2] P. Koopman, "Embedded System Design Issues- The Rest of the Story", Proceedings of International Conference on Computer Design, Austin, 7-9 October, 1996, pp.

[3] P. Koopman, "Design Constraints on Embedded Real Time Control Systems", System Design and Network Architecture Conference, May 8-10, 1990, pp. 71-77.

[4] L. Khelladi, Y. Challal, A. Bouabdallah, N. Badache, "On Security Issues in Embedded Systems: Challenges and Solutions", International Journal of Information and Computer Security 2008, Vol. 2, No.2, pp. 140-174.

[5] R. Tahir, K. D. McDonald Maier, "Improving Resilience against Node Capture Attacks in Wireless Sensor Networks using ICMetrics", IEEE Conference on Emerging Security Technologies, Portugal, September 5-7, 2012.

[6] C. Henri, "A Course in Computational Algebraic Number Theory", Graduate Texts in Mathematics, Berlin, New York.

[7] G. Shmonin, "Hermite normal form: Computation and applications", Notes available at http://disopt.epfl.ch/webdav/site/disopt/shared/IntPoints2009/hnf.pdf, February 24, 2009.

[8] E. Papoutsis, W. G. J. Howells, A. B. T. Hopkins, and K. D. McDonald-Maier, "Integrating Multi-Modal Circuit Features within an Efficient Encryption System", Third International Symposium on Information Assurance and Security, IEEE Computer Society Washington, DC, USA, 2007, pp. 83-88.

[9] E. Papoutsis, W. G. J. Howells, A. B. T. Hopkins, and K. D. McDonald-Maier, "Ensuring Secure Healthcare Communications via ICmetric based Encryption on unseen Devices", Symposium on Bio-inspired, Learning and Intelligent Systems for Security, Edinburgh, 20-21 Aug. 2009, pp. 113-117.

[10] E. Papoutsis, W. G. J. Howells, A. B. T. Hopkins, and K. D. McDonald-Maier, "Integrating Feature Values for Key Generation in an ICmetric System," in IEEE NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2009) San Francisco, California, 2009, pp. 82-88

[11] E. Papoutsis, W. G. J. Howells, A. B. T. Hopkins, and K. D. McDonald-Maier, "Ensuring Secure Healthcare Communications via ICmetric based Encryption on unseen Devices", Symposium on Bio-inspired, Learning and Intelligent Systems for Security, Edinburgh, 20-21 Aug. 2009, pp. 113-117.

[12] E. Papoutsis, W. G. J. Howells, A. B. T. Hopkins, and K. D. McDonald-Maier, "Integrating Feature Values for Key Generation in an ICmetric System," in IEEE NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2009) San Francisco, California, 2009, pp. 82-88.

[13] E. Papoutsis, W. G. J. Howells, A. B. T. Hopkins, and K. D. McDonald-Maier, "Key Generation for Secure Inter-satellite Communication," in IEEE, NASA/ESA Conference on Adaptive hardware and Systems 2007, AHS-2007 Edinburgh, UK, 2007, pp. 671-681

[14] R. Tahir, K. D. McDonald Maier, "An ICMetrics based Lightweight Security Architecture using Lattice Signcryption", IEEE Conference on Emerging Security Technologies, Portugal, September 5-7, 2012.

[15] G. O. Karame, S. Capkun, U. Maurer , "Privacy-Preserving Outsourcing of Brute-Force Key Searches", Proceedings of the 3rd ACM workshop on Cloud Computing Security, New York, USA, pp.101-112.

[16] F. F. Yao, Y. L. Yin, "Design and Analysis of Password-Based Key Derivation Functions", IEEE Transactions on Information Theory, Vol 51(9), pp. 3292-3297.

[17] B. Kaliski, "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, Sept. 2000.

[18] C. Adams, G. Kramer, S. Mister and R. Zuccherato, “On the Security of Key Derivation Functions”, Industrial Simulation Conference (ISC’2004), Spain, pp. 134-145.

[19] J. Kelsey, B. Schneier, C. Hall, and D. Wagner, “Secure Applications of Low-Entropy Keys”, Information Security Workshop (ISW 1997), Japan, pp. 121-134.

[20] NIST's Policy on Hash Functions, National Institute on Standards and Technology Computer Security Resource Center, March 29, 2009.

[21] C. Paar, J. Pelzl, “Hash Functions-Understanding Cryptography, A Textbook for Students and Practitioners", Springer, 2009.

[22] X. Wang, Y. L. Yin, and H. Yu, “Finding Collisions in the Full SHA-1”, 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, pp. 17-36.

[23] B. Schneier, "Schneier on Security: Cryptanalysis of SHA-1", Schneier.com.

[24] “Secure Hash Standard (SHS)”, FIPS PUB 180-3, October 2008, http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf.

[25] A. A. Pirzada, C. McDonald, “Kerberos Assisted Authentication in Mobile Ad hoc Networks” Proceedings of the 27th conference on Australasian Computer Science, Australia, Vol 26, pp. 41-46.

[26] J. Clark, J. Jacob, “A Survey of Authentication Protocol Literature: Version 1.0 17”, November 1997.

[27] J. T. Kohl, B.C. Neuman, “The Kerberos Network Authentication Service”, RFC1510, 1993.

[28] D. Harkins, D. Carrel, “The Internet Key Exchange (IKE)”, RFC2409, 1998.

[29] B. Schneier, “Applied Cryptography: Protocols, Algorithm, and Source Code in C”, John Wiley & Sons, Inc., 1996.

[30] X. Li, J. Han, Z. Sun, “Design Principles and Security of Authentication Protocols with Trusted Third Party”, AUUG 2004, Australia pp. 103-107.