Reconfigurable Implementation of Hierarchical Identity Based Encryption for Wireless Sensor Networks

A.Praveena,
Assistant Professor, Department of Information Technology, Dr.N.G.P Institute of Technology, Coimbatore,

Abstract- Once considered a playground for hackers and malicious attacks, wireless networks are fast becoming more secure than their wired counterparts. Developments in micro electro mechanical systems (MEMS) and wireless networks are opening a new domain in networking history. Recent technological advances in wireless networking, IC fabrication and sensor technology have lead to the emergence of millimetre scale devices that collectively form a Wireless Sensor Network (WSN) and are radically changing the way in which we sense, process and transport signals of interest. They are increasingly become viable solutions to many challenging problems and will successively be deployed in many areas in the future such as in environmental monitoring, business, and military applications. The huge challenge in WSN is due to inherent resource and computing constraints. Because the sensor nodes are battery powered, increasing the autonomous lifetime of a WSN is a challenging optimization problem.

However, deploying new technology, without security in mind has often proved to be unreasonably dangerous. There have been significant contributions to overcome any weaknesses in sensor networks like coverage problems, lack in power and making best use of limited network bandwidth, however; work in sensor network security is still in its infancy stage. The problem of securing these networks emerges more and more as a hot topic. Symmetric key cryptography is commonly seen as infeasible and public key cryptography has its own key distribution problem. In contrast to this prejudice, this paper presents a method to increase the lifetime of a WSN by minimizing the energy cost of transporting information from a set of sources nodes to the sink nodes and for achieving security we have used a new public-key encryption technology called hierarchical identity-based encryption (HIBE) which allows to calculate a public key directly from a user's identity. By calculating public keys instead of generating them randomly, many of the difficulties that make encryption technology difficult to deploy and maintain are eliminated, making encrypted communications much easier to implement than in the past.

Index Terms: Wireless Sensor Networks, Cryptography, Hierarchical Identity Based Encryption.

I. INTRODUCTION

RECENT advancements in the design and fabrication of low power VLSI circuitry, along with wireless communications, have broadened the applications prospects for wireless sensor networks. These networks are quickly gaining popularity due to the fact that they are potentially low cost solutions to a variety of real world challenges and are expected to play an essential role in the upcoming age of pervasive computing.

Sensor network s are given by a large number of sensor nodes that are densely deployed either inside or close to a phenomenon of interest with computational capabilities connected through wireless links. Each sensor node is an independent, low-power, smart device with sensing, processing and wireless communication capabilities. From national defense, medical applications, to the environment, the data delivered from the sensor networks are unstructured, using their own format and protocols. Sensor networks are delivering near-real-time information to scientists worldwide. Extracting this information to gain knowledge and understanding is one of the greatest challenges faced today.

Sensor networks are dense wireless networks of small, low-cost sensors, which collect and disseminate environmental data. These networks are an important ingredient of "anywhere and anytime" ubiquitous wireless next generation communication infrastructure. WSN is a combination of nodes that are used to sense data from its environment and to send the aggregated data to its control node often called sink. In this diversified yet integrated future network environments, sensor network has a role of reliable monitoring and control of variety of applications based on environmental sensing. They have applications in a variety of fields such as environment monitoring which involves monitoring air, soil and water, condition based maintenance, habitat monitoring (determining the plant and animal species population and behavior), seismic detection, military surveillance, inventory tracking, smart spaces and gathering sensing information in inhospitable locations, medical and home security to machine diagnosis, chemical/biological detection etc.

Wireless sensor networks facilitate monitoring and controlling of physical environments from remote locations with better accuracy. In spite of the diverse applications, sensor networks pose a number of unique technical challenges due to the following factors: Adhoc deployment, unattended operation, untethered, and dynamic changes. In this paper for achieving security we have used a new public-key encryption technology called hierarchical identity-based encryption (HIBE) and we have used Dual HIDE for achieving better security.

A. Contributions of the paper

This paper is intended to be an introduction to wireless sensor networks—with an emphasis on structural and environmental monitoring applications— for engineers and scientist unfamiliar with the computer science and engineering concepts. We try to give a thorough but general survey of the area and refer to several papers in the computer science and engineering literature for more detailed information. In this paper for achieving security we have used a new public-key encryption technology called hierarchical identity-based encryption (HIBE). Hierarchical identity-based encryption (HIBE) schemes were proposed to alleviate the workload of a root PKG by delegating private key generation and identity authentication to lower-level PKGs. The organization of PKGs and users forms a hierarchy that is rooted by the root PKG.

The rest of the paper is described as follows. Section 2 discusses the background information for the architecture of WSN and components of a sensor node. The motivation for the proposed scheme presented is discussed in Section 3. Section 4 discusses the related work. Section 5 discusses the limitations with the previous work. Section 6 discusses the proposed scheme followed by conclusions and future work.

II. SENSOR NETWORK ARCHITECTURE

In the typical architecture of WSN the sensor nodes are usually scattered in a sensor field. Each of these scattered sensor nodes has the capabilities to collect data and perform partial or no processing on the data. Each sensor node has the required infrastructure to communicate with the other nodes. Data are routed back tot the sink/base station by a multihop infrastructure less architecture through the sink. We will distinguish a special type of node called a gateway node. Gateway nodes are connected to components outside of the sensor network through long range communication (such as cables or satellite links), and all communication with users of the sensor network goes through the gateway node.

The sink node communicates with the task manager via core network which can be Internet or Satellite. Since Sensors are low cost, low power, and small in size, the transmission power of a sensor is limited. The data transmitted by a node in the field may pass through multiple hops before reaching the sink. Many route discovery protocols (mostly inherited from Ad hoc networks) have been suggested for maintaining routes from field sensors to the sink(s). Due to low memory, scarcity of available bandwidth and low power of the sensors, many researchers considered these separate route discovery mechanisms undesirable.

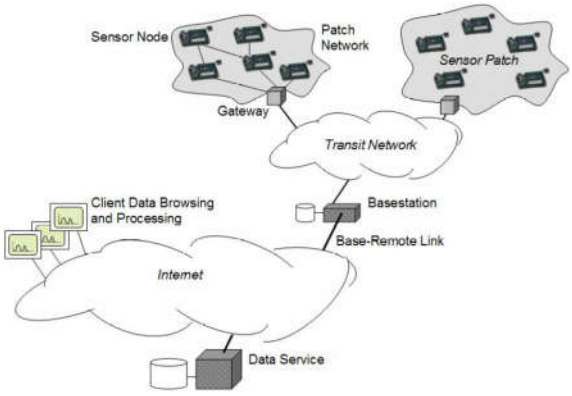


Fig 1. Typical Sensor Network

Once sensors are deployed they remain unattended, hence all operations e.g. topology management, data management etc. should be automatic and should not require external assistance. In order to increase the network life time, the communication protocols need to be optimized for energy consumption. It means a node must be presented lowest possible data traffic to process.

The figure 2 shows the components of a sensor node. A sensor node is made up of four basic components: a sensing unit, a processing unit, a transceiver unit and a power unit. They may also have additional application-dependent components such as a location finding system, power generator and mobilizer. Sensing units are usually composed of two subunits: sensors and analog to digital converter. The analog signals produced by the sensors based on the observed phenomenon are converted to digital signals by the ADC, and then fed to the processing unit. The processing unit is generally associated with a small range a small storage unit, manages the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of the sensor network is the power unit. Power unit may be supported by power scavenging units such as solar cells. There are also other subunits that are application dependent.

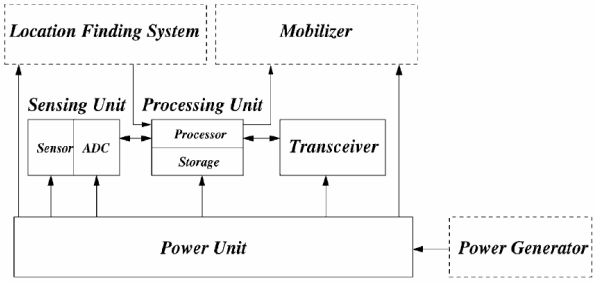


Fig 2. Components of a sensor node

The emergence of sensor networks as one of the dominant technology trends in the coming decades has posed numerous unique challenges to researchers. These networks are likely to be composed of hundreds, and potentially thousands of tiny sensor nodes, functioning autonomously, and in many cases, without access to renewable energy resources. Cost constraints and the need for ubiquitous, invisible deployments will result in small sized, resource-constrained sensor nodes. Such sensor nodes have the following resource constraints:

Communication: The wireless network connecting the sensor nodes provides usually only a very limited quality of service has latency with high variance, limited bandwidth, and frequently drops packets.
Power consumption: Sensor nodes have limited supply of energy, and thus energy conservation needs to be of the main system design considerations of any sensor network application. For example, the MICA motes are powered by two AA batteries that provide about 2000mAh powering the mote for approximately one year in the idle state and for one week under full load.
Computation: Sensor nodes have limited computing power and memory sizes. This restricts the types of data processing algorithms on a sensor node, and it restricts the sizes of intermediate results that can be stored on the sensor nodes.
Uncertainty in sensor readings: Signals detected at physical sensors have inherent uncertainty, and they may contain noise from the environment. Sensor malfunction might generate inaccurate data, and unfortunate sensor placement (such as a temperature sensor directly next to the air conditioner) might bias individual readings.
While the set of challenges in sensor networks are diverse, we focus on fundamental security challenges in this paper.

III. SECURITY ISSUES IN SENSOR NETWORKS

Because the sensor nodes are battery powered, increasing the autonomous lifetime of a WSN is a challenging optimization problem. Communication of data within a WSN is one of the most energy-expensive tasks a node undertakes – using data compression to reduce the number of bits sent reduces energy expended for communication. Data compression which highly reduces the communication overhead by aggregating and compressing data packets can be performed at intermediate sensor nodes. However, compression requires computation, which also expends energy. Fortunately, trading computation for communication can save energy since a recent paper1 asserts that typically on the order of 3000 instructions can be executed for the energy cost required to communicate one bit over a distance of 100 m by radio.

Apart from achieving energy efficiency many WSN applications that span military and civilian use assume that the sensor nodes will be deployed hostile environments and thus be prone to a wide variety of malicious attacks. As a result, security becomes a key concern. Sensor networks are particularly vulnerable to several key types of attacks, such as denial of service attacks, traffic analysis, privacy violation, physical attacks, node take overs, attacks on routing protocols, etc.

The data transported and exchanged between sensor nodes is critical. Such data has to be protected against threats in a way so classic security properties like integrity, authenticity or confidentiality can be guaranteed[12].To accomplish such security goals in modern networks like the Internet or companies local area network cryptographic primitives like encryption / decryption as well as signature schemes are usually needed. Keys for encryption purposes must be agreed upon by communicating nodes. Due to resource constraints, achieving such key agreement in wireless sensor networks is non-trivial. Many key agreement schemes used in general networks, such as Diffie-Hellman and public-key based schemes, are not suitable for wireless sensor networks. Pre-distribution of secret keys for all pairs of nodes is not viable due to the large amount of memory used when the network size is large.

The lack of a fixed infrastructure and the ad hoc nature of WSN deployments suggest that the ability to encrypt and decrypt confidential data among arbitrary sensor nodes while enabling undisputed authentication of all parties will be a fundamental prerequisite for achieving security. To do this, nodes must be able to establish a secret key and know who their counterparts are. Thus, it becomes highly desirable to have a secure and efficient distribution mechanism that allows simple key generation for large-scale sensor networks while facilitating all the necessary authentications.

Although a variety of key-generation methods have been developed, they cannot be directly applied to sensor network environments due to the problems such as very limited resources (memory, power), unreliable communication (unreliable transfer, conflicts, latency), Unattended Operation (Exposure to Physical Attacks, Managed Remotely, No Central Management Point) etc. Due to these constraints it is difficult to directly employ the existing security approaches to the area of wireless sensor networks.

IV. Previous Work

Because of the problems mentioned in previous section security is commonly considered as a delicate problem. One security aspect that receives a great deal of attention in WSN is the area of key management. The two possibilities for achieving security is to use symmetric cryptography and public key cryptography. Two of the major techniques used to implement public-key cryptosystems are RSA and elliptic curve cryptography (ECC).But most security work on WSN focuses on the search for and development of alternatives to classical public-key algorithms and public key infrastructures. Recent work has challenged notion that Diffie-Hellman and public key based schemes are infeasible in WSNs. Recently; however, several groups have successfully implemented public-key cryptography (to varying degrees) in wireless sensor networks.

Researches have demonstrated that basic ECC key generation can in fact be attained sensor nodes in reasonable time and with predictable improved performance. ECC has thus emerged as a suitable public key cryptographic foundation that provides high security for relatively small key sizes. In [1] Gura et al. report that both RSA and elliptic curve cryptography are possible using 8-bit CPUs with ECC demonstrating a performance advantage over RSA. Another advantage is that ECC's 160 bit keys result in shorter messages during transmission compared the 1024 bit RSA keys. In particular Gura et al. demonstrate that the point multiplication operations in ECC are an order of magnitude faster than private-key operations within RSA, and are comparable (though somewhat slower) to the RSA public-key operation [1].

In [3], Watro et al. show that portions of the RSA cryptosystem can be successfully applied to actual wireless sensors, specifically the UC Berkeley MICA2 motes [2]. In particular, they implemented the public operations on the sensors themselves while offloading the private operations to devices better suited for the larger computational tasks. Compared to RSA, the prevalent public-key scheme of the Internet today, Elliptic Curve Cryptography (ECC) offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings and is thus better suited for small devices.

Shamir [17] proposed the idea of identity-based cryptography in 1984, and described an identity-based signature scheme in the same article. However, practical Identity-based encryption (IBE) schemes were not found until recently with the work of Boneh and Franklin [5,6] and Cocks [8] in 2001. Cocks's scheme is based on the Quadratic Residuosity Problem, and although encryption and decryption are reasonably fast (about the speed of RSA), there is significant message expansion, i.e., the bit-length of the cipher text is many times the bit-length of the plaintext. The Boneh-Franklin scheme bases its security on the Bilinear Diffie-Hellman Problem, and is quite fast and efficient when using Weil or Tate pairings on super singular elliptic curves or abelian varieties. We must note that ID-based encryption has some disadvantages. Bob receives his private key from a third party called a Private Key Generator (PKG) that computes his private key as a function of its master secret and Bob's identity. This requires Bob to authenticate himself to the PKG (in the same way he would authenticate himself to a CA), and requires a secure channel through which the PKG may send Bob his private key.

Bob's PKG must publish parameters that embed its master secret, and Alice must obtain these parameters before sending an encrypted message to Bob. Another disadvantage is that the PKG knows Bob's private key, i.e., key escrow is inherent in ID-based systems. Clearly, escrow is a serious problem for some applications. However, the advantages of identity-based encryption are compelling. The problem of obtaining authentic public keys has been replaced by the problem of obtaining authentic public parameters of PKGs, but the latter should be less burdensome since there will be substantially fewer PKGs than total users. For example, if everyone uses a single PKG, then everyone in the system can communicate securely without ever having to perform online lookup of public keys or public parameters.

V. Limitations with the Previous Work

Symmetric cryptography, which is computationally inexpensive, can be used to achieve some of these security goals. One major drawback with this approach is the key exchange problem i.e. the two communication nodes must somehow know the shared key before they can communicate securely.

So the problem that arises is how to ensure that the shared key is indeed shared between the two hosts who wish to communicate and no other rogue hosts who may wish to eavesdrop. How to distribute a shared key securely to communicating hosts is a non-trivial problem since pre-distributing the keys is not always feasible. Unfortunately, capturing even a single node, in the network would easily reveal the network's secret key. So it is inflexible with respect to key management as it requires pre-distribution of keys. On the other hand, public key cryptography allows for flexible key management, but requires a significant amount of computation.

The main difficulty today in developing secure systems based on public key cryptography is not the problem of choosing appropriately secure algorithms or implementing those algorithms. Rather, it is the deployment and management of infrastructures to support the authenticity of cryptographic keys: there is a need to provide an assurance to the user about the relationship between a public key and the identity (or authority) of the holder of the corresponding private key. In a traditional Public Key Infrastructure (PKI), this assurance is delivered in the form of certificate, essentially a signature by a Certification Authority (CA) on a public key [1].The issues associated with certificate management, including revocation, storage and distribution and the computational cost of certificate verification. These are particularly acute in processor or bandwidth-limited environments.

In 1984, Shamir [31] proposed a concept of Identity-based cryptography where users' identifier information such as email or IP addresses instead of digital certificates can be used as public key for encryption or signature verification. As a result, identity-based cryptography significantly reduces the system complexity and the cost for establishing and managing the public key authentication framework known as Public Key Infrastructure (PKI). In IBE schemes private key generator (PKG) is responsible for generating private keys for all users, and therefore is a performance bottleneck for organizations with large number of users.

VI. Proposed Work

To mitigate the damage caused by the exposure of secret key information in IBE, one way is to construct a hierarchical identity based encryption. Below, we discuss the notion of HIBE in more detail.

A. Motivation for Hierarchical Identity Based Encryption

Although having a single PKG would completely eliminate online lookup, it is undesirable for a large network because the PKG becomes a bottleneck. Not only is private key generation computationally expensive, but also the PKG must verify proofs of identity and must establish secure channels to transmit private keys. Hierarchical ID-based encryption (HIDE) allows a root PKG to distribute the workload by delegating private key generation and identity authentication to lower-level PKGs. In a HIDE scheme, a root PKG need only generate private keys for domain-level PKGs, who in turn generate private keys for users in their domains in the next level. Authentication and private key transmission can be done locally. To encrypt a message to Bob, Alice need only obtain the public parameters of Bob's root PKG (and Bob's identifying information); there are no "lower-level parameters." Another advantage of HIDE schemes is damage control: disclosure of a domain PKG's secret does not compromise the secrets of higher-level PKGs. The schemes of Cocks and Boneh-Franklin do not have these properties.

A hierarchical ID-based key sharing scheme with partial collusion-resistance is given in [10,11]. Horwitz and Lynn [12] introduced hierarchical identity-based encryption, and proposed a 2-level HIDE scheme with total collusion-resistance at the first level and with partial collusion-resistance at the second level, i.e., (a threshold number of) users can collude to obtain the secret of their domain PKG (and thereafter masquerade as the domain PKG). This scheme may be practical for applications where collusion below the first level is not a concern. Finding a secure and practical hierarchical identity-based encryption scheme was, prior to this paper, an important open question.

B. Preliminaries

Hierarchical Identity-Based Encryption (HIDE): A HIDE scheme is specified by five randomized algorithms: Root Setup, Lower-level Setup, Extraction, Encryption, and Decryption. The security of our HIDE scheme is based on the difficulty of the Bilinear Diffie-Hellman (BDH) Problem. Let G1 and G2 be two cyclic groups of some large prime order q. We write G1 additively and G2 multiplicatively. Our HIDE scheme makes use of a "bilinear" pairing.

BDH Parameter Generator: As in [5], we say that a randomized algorithm IG is a BDH parameter generator if IG takes a security parameter $K > 0$, runs in time polynomial in K, and outputs the description of two groups G1 and G2 of the same prime order q and the description of an admissible pairing $\hat{e}$: G1x G1 → G2.

Bilinear Diffie-Hellman (BDH) Problem: Given a randomly chosen $P \in$ G1, as well as aP, bP, and cP (for unknown randomly chosen $a, b, c \in Z/qZ$), compute $\hat{e}(P, P)^{abc}$. For the BDH problem to be hard, G1 and G2 must be chosen so that there is no known algorithm for efficiently solving the Diffie-Hellman problem in either G1 or G2. Note that if the BDH problem is hard for a pairing $\hat{e}$, then it follows that $\hat{e}$ is non-degenerate.

C. Choosing the HIBE Key in WSN

In sensor networks, attributes such as location identify the final traffic destination and are even used directly by the routing protocol instead of a network address [4]. The reason is that more common attributes can be encoded in only a few bits. Each node still has a unique network address, but only very rarely is this used for routing. Each node has a network-wide unique ID and a low-power transceiver. Its range may differ due to variations in device implementation and wireless propagation environment; such that communication links between two nodes are not necessarily bidirectional. So the network-wide unique ID of sensor node can be chosen as HIBE key for encryption.

D.HIBE System Components

Let Level$_i$ be the set of entities at level i, where Level$_0$ = {Root Public Key Generator (PKG)}. Let K be the security parameter given to the setup algorithm, and let IG be a BDH parameter generator.

Root Setup: The root PKG

1. Runs IG on input K to generate groups G1, G2 of some prime order q and an admissible pairing $\hat{e}$:G1xG1→ G2;
2. Chooses an arbitrary generator $P_0 \in G1$;
3. Picks a random $s_0 \in Z/qZ$ and sets $Q_0 = s_0P_0$;
4. chooses cryptographic hash functions $H_1 : \{0, 1\} \rightarrow$ G1 and H_2 : G2 →$\{0, 1\}^n$ for some n. The security analysis will treat H_1 and H_2 as random oracles. The message space is $M = \{0, 1\}^n$. The cipher text space is $C = G^t_1$ x $\{0, 1\}^n$ where t is the level of the recipient. The system parameters are $params$ = (G1, G2, $\hat{e}$, P_0, Q_0, H_1, H_2). The root PKG's secret is $s0 \in Z/qZ$.

Lower-Level Setup: Entity $E_t \in$ Level$_t$ picks a random $s_t \in Z/qZ$, which it keeps secret.

Extraction: Let E_t be an entity in Level$t$ with ID-tuple (ID$_1$, . . . , ID$_t$), where (ID$_1$, . . . , ID$_i$) for $1 \le i \le t$ is the ID-tuple of E_t's ancestor at Level$_i$. Set S_0 to be the identity element of G1. Then E_t's parent:

1. Computes $P_t = H_1$(ID$_1$, . . . , ID$_t$) $\in$ G1;
2. Sets E_t's secret point S_t to be $S_{t\ 1} + s_{t\ 1}P_t = \sum^{t}_{i=1} s_{i\ 1}P_i$;
3. Also gives E_t the values of $Q_{i} = s_iP_0$ for $1 \le i \le t$ 1.

Encryption: To encrypt $M \in M$ with the ID-tuple (ID$_1$, . . . , ID$_t$), do the following:

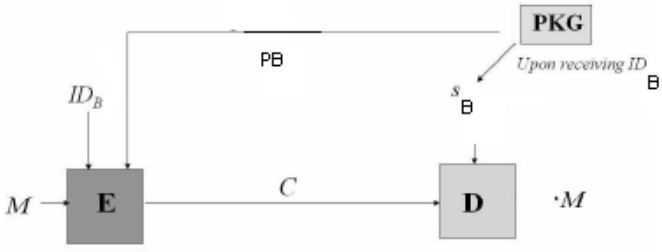
1. Compute $P_i = H_1$(ID$_1$, . . . , ID$_i$) $\in$ G1 for $1 \le i \le t$.
2. Choose a random $r \in Z/qZ$ and set the cipher text as
$C = [rP_0, rP_2. . , rP_t, M \in H_2(g^r)]$where $g=\hat{e}(Q_0,P_1) \in$ G2.

Decryption: Let $C = [U_0, U_2, . . . , U_t, V] \in C$ be the cipher text encrypted using the ID-tuple (ID$_1$, . . . , ID$_t$). To decrypt C, E_t computes:

$$V \text{ xor } H_2(\hat{e}(U_0, S_t)/\Pi^{t}_{i=2} \hat{e}(Q_{i\ 1}, U_i)) = M.$$

E. Architecture

1. A root TA (Trusted Authority PKG) at level 0 with a master secret s0.
2. Entity at level t-1 in hierarchy has secret st-1 and issues private keys St to entities at level t for which it is responsible.
3. So each entity acts as TA for lower-level entities.
4. Any entity can encrypt for (or verify signatures of) any other entity in the hierarchy, provided their identity string is known.



E: Identity Based Encryption D: Identity Based Decryption
M: Sensor Data C. Encrypted Data
P_B: Public Key computed for Encryption
S_B Secnet Key computed
ID_B Identifier of receiving Sensor node B

Fig 3: IBE Encryption System in WSN

VII. VHDL BASED IMPLEMENTATION

The design is implemented using the VHDL language, the Xilinx Synthesis Technology (XST) tools, and a Spartan IIE 1.8V FPGA platform. Hardware description languages are being increasingly popular in designing large scale integrated circuits. Some of the popular HDL's are VHDL (VHSIC Hardware Description Language where VHSIC-Very High Speed Integrated Circuits) and VerilogHDL. VHDL is a popular HDL that can be used to model a digital system at many levels of abstraction, ranging from algorithmic level to the gate level and can also be described hierarchically. It supports many of the features in high level languages. The fundamental motivation to use VHDL is that it is a standard, technology/vendor independent language, and is therefore portable, reusable and promotes rapid prototyping. Therefore the vital advantage is its device independent nature. The designer's source code can be targeted to any technology without changes which provides reduced design cycle times, faster time to market and reduced cost.

Once the VHDL code has been written, it can be used either to implement the circuit in programmable device (from Altera, Xilinx, Atmel) or can be submitted to foundry for fabrication of ASIC chip.

A. FPGA Design Methodology

The target FPGA device was Xilinx SpartanIIE XC2S200E. All algorithms were first described in VHDL, and their description verified through the functional simulation using ModelSim XE II v5.7c, a simulator from Mentor Graphics Company. Test vectors and intermediate results from the reference software implementations based on Crypto++ library [1] were used for debugging and verification of VHDL codes.

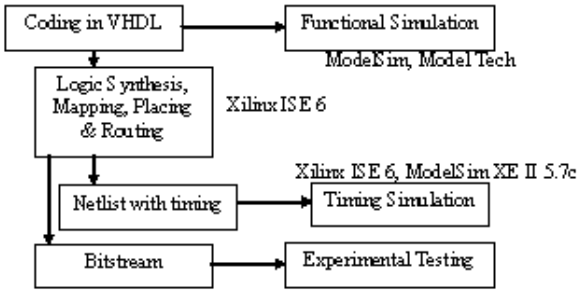


Fig 4: Design Flow & Tools in Development of Cryptographic Modules

The revised VHDL code became an input to the Xilinx ISE 6, performing the automated logic synthesis, mapping, placing, and routing. Tools included in this environment generated reports describing the area and speed of implementation, a net list used for timing simulation, and a bit stream used to configure an actual FPGA device. This newer simulator (ISE + ModelSim) offers a much broader set of features, which allow, a more refined timing analysis. All designs were fully verified through behavioral, post-synthesis, and timing simulations, and experimentally tested. The Bit stream (stored for production solution in DPRAM) that is transferred contains all information to define the logic and interconnect of the design and is different for every design.

B. Advantages of FPGA Based Implementation

Due to the comparative slowness of the public key crypto algorithms, dedicated hardware support is desirable. Reconfigurable hardware devices such as FPGAs are an appealing alternative for the implementation of cryptographic algorithms. Their advantages combine flexibility and ease of upgrade (modification of software) with improved physical security and performance. In addition, the time and cost of FPGA design are smaller than in other hardware approaches (ASIC) which has longer design cycle. These capabilities of FPGAs make them a suitable platform for cryptographic applications. Their structure allows complex arithmetic operations that are not suited to general purpose CPUs to be implemented more efficiently. The fast prototyping development time of an FPGA design allows modifications to be implemented with relative ease. Also, the newest generation of FPGA devices, features very sophisticated internal architectures help designers to make better use of available resources. Though software implementations of hash functions provide ease of use, ease of upgrading, portability, flexibility, hardware implementation has more physical security by nature, as it can not easily be modified by an attacker. But speed of software implementation is restricted to speed of computing platform and there are vulnerabilities for viruses and other complications due to system failures.

VIII. SECURITY ANALYSIS OF HIBE SYSTEM

A. Secure Against Key Escrow Problem

In IBE schemes, key escrow is “inherent” because the PKG knows the private key of each user. Even in the hierarchical scheme of Horwitz and Lynn, every ancestor of a given user in the hierarchy knows that user’s private key. Although this key escrow property may be useful in some contexts, it is certainly not desirable for all applications. In our HIDE scheme, since the private point of a user depends on a secret number known only to the parent of that user, no ancestor other than the parent may compute the user’s particular private point. However, the user’s ancestors can still decrypt the user’s mail; they may simply compute a different (but equally effective) private key for the user based on different lower-level Q-values. Using these different Q-values, they may also forge the user’s signature. In this section, we discuss how Dual-HIDE and/or key agreement protocols can be used to restrict this key escrow property.

Dual HIDE: In 2000, Sakai, Ohgishi and Kasahara [16] presented a “key sharing scheme” based on the Weil pairing. The idea was quite simple: suppose a PKG has a master secret s, and it issues private keys to users of the form sP_y, where $P_y = H_1(\mathrm{ID}_y)$ and ID_y is the ID of user y(as in Boneh-Franklin). Then users y and z have a shared secret that only they (and the PKG) may compute, namely, $\hat{e}(sP_y, P_z) = \hat{e}(P_y, P_z)s = \hat{e}(P_y, sP_z)$. They may use this shared secret to encrypt their communications.

Notice that this key sharing scheme does not require any interaction between the parties. We can view Sakai, Ohgishi and Kasahara's discovery as a type of "dual-identity-based encryption," where the word "dual" indicates that the identities of the sender and the recipient (rather than just the recipient) are required as input into the encryption and decryption algorithms. The main practical difference between this scheme and the Boneh- Franklin IBE scheme is that the sender must obtain its *private key* from the PKG before sending encrypted communications, as opposed to merely obtaining the *public parameters* of the PKG.

In the hierarchical context, Dual-HIDE may be more efficient than HIDE if the sender and recipient are close to each other in the hierarchy tree. Suppose two users, y and z, have the ID-tuples $(ID_{y1}, \ldots, ID_{yl}, \ldots, ID_{ym})$ and $(ID_{z1}, \ldots, ID_{zl}, \ldots, ID_{zn})$, where $(ID_{y1}, \ldots, ID_{yl}) = (ID_{z1}, \ldots, ID_{zl})$. In other words, user y is in $Level_m$, user z is in $Level_n$, and they share a common ancestor in $Level_l$.

Restricting Key Escrow using Dual HIDE: Consider again users y and z that have a common ancestor in $Level_l$. Let's say their common ancestor is Cryptography State University, and suppose that user y uses Dual-HIDE to encrypt its messages to z. As stated above, CSU's parent knows CSU's private point. From CSU's perspective, this may be an undesirable situation. However, CSU can easily change its private point S_l by setting $S_l := S_l + bP_l$ and setting $Q_{l\ 1} := Q_{l\ 1} + bP_0$ for some random $b \in Z/qZ$. This new private key is just as effective, and is unknown to CSU's parent. Assuming that CSU uses its new private key to issue private keys to its children, none of CSU's ancestors will be able to decrypt y's message to z encrypted using Dual-HIDE. More specifically, only ancestors of z that are within CSU's domain will be able to decrypt.

B. Secure Against Chosen Ciphertext Attack

We say that a HIDE scheme is semantically secure against adaptive chosen cipher text and adaptive (resp., non adaptive) chosen target attack (IND-HIDCCA (resp. IND-NHID-CCA)) if no polynomially bounded adversary A has a non-negligible advantage against the challenger in the following game.

IX. FUTURE ENHANCEMENTS

Improving Efficiency of Encryption: Levels 0 and 1 can be merged into a single (combined levels 0 and 1) root PKG. In that case, $g = \hat{}e(Q_0, P_1)$ is included in the system parameters. This saves encrypters the task of computing the value of this pairing. However, decrypters must compute an extra pairing (as a result of being one level lower down the tree).

Distributed PKGs: As in Section 6 of [6], the secrets s_i and private keys can be distributed using techniques of threshold cryptography to protect the secrets and make the scheme robust against dishonest PKGs.

Concrete Schemes: For our HIDE schemes, one can use the same elliptic curves or abelian varieties.

X. CONCLUSION

As the applications of wireless sensor networks tend to increase more rapidly, the problem of achieving energy efficient communication and securing them against attacks becomes much more important. Without proper security, it is impossible to completely trust the results reported from sensor networks deployed outside of controlled environments. In this paper we have seen how one can use the Hierarchical Identity Based Encryption achieve secure communication in WSN. We gave hierarchical ID-based encryption (HIDE) schemes that are practical, totally collusion-resistant, and secure against chosen-cipher text attacks. The message expansion factor and complexity of decryption grow only linearly with the number of levels in the hierarchy. We used a related Dual hierarchical ID-based scheme that is especially effective when used in combination with HIDE. Many of the difficulties that make Public key encryption technology difficult to deploy and maintain are eliminated, making encrypted communications much easier to implement than in the past.

REFERENCES

[1] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz. Comparing elliptic curve cryptography and RSA on 8-bit cpus. In 2004 workshop on Cryptographic Hardware and Embedded Systems, Aug. 2004.
[2] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus. Tinypk: Securing sensor networks with public key technology. Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN '04), pp. 59–64. ACM Press, 2004.
[3] D. CopperSmith, "Fast evolution of algorithms in fields of characteristic row ", IEEE Transactions on Information Theory, 30 (1984), 587-594.
[4] W. Dillie and V.Hellman, "New Directions in Cryptography ", IEEE Transactions on Information Theory, 22 (1976), 644-654.
[5] J. Pollard. Monte Carlo Methods for Index Computation (mod p). Mathematics of Computation, 32:918–924, 1978.
[6] D. Shanks. A Theory of Factorization and Genera. In Proc. Symp. Pure Math., 20:415–440, 1971.
[8] D. G. Cantor. On the analogue of the division polynomials for hyper elliptic curves. Math., 447:91–145, 1994.
[7] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. Commun. ACM 47(6):53–57, 2004.

[8]. Menezes, A., Okamoto, T., and Vanstone, S. "Reducing elliptic curve logarithms to logarithms in a finite field". proceedings of the twenty-third annual ACM symposium on Theory of computing. Annual ACM Symposium on Theory of Computing. ACM Press, 1991: p 80 – 89.
[9].S. AlRiyami and K.G. Paterson. Certificateless public key cryptography. In Advances in Cryptology – ASIACRYPT 2003, vol. 2894 of LNCS, pp. 452–473, 2003.
[10]. D. Boneh and M. Franklin. Identity-Based encryption from the Weil pairing. SIAM Journal of Computing, 32(3):586–615, 2003. This is the full version of an extended abstract of the same title presented at Crypto'01.
[11]. C. Cocks, An Identity Based Encryption Scheme Based on Quadratic Residues, Proceedings of the 8th IMA International Conference on Cryptography and Coding, LNCS 2260, pages 360-363, Springer-Verlag, 2001.
[12].D. Boneh and X. Boyen, Secure Identity Based Encryption without Random Oracles, extended abstract in Proceedings of CRYPTO '04, LNCS 3152, Springer-Verlag, 2004. Full paper available in the IACR eprint archives.
[13].Stallings, W. Cryptography and Network Security. Prentice Hall, 2003.
*[14] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing, in Advances in Cryptology – Crypto 2001, Lecture Notes in Computer Science **2139** (2001), Springer, 213–229.*
*[15]G. Hanaoka, T. Nishioka, Y. Zheng, and H. Imai. An efficient [hierarchical identity based key-sharing method resistant against collusion-attacks, in Advances in Cryptology – Asiacrypt 1999, Lecture Notes in Computer Science **1716** (1999), Springer, 348–362.*
*[16] J. Horwitz and B. Lynn. Toward Hierarchical Identity-Based Encryption, in Advances in Cryptology – Eurocrypt 2002, Lecture Notes in Computer Science **2332** (2002), Springer, 466–481.*