RC24608 (W0807-108) July 17, 2008
Computer Science

IBM Research Report

Trust Management for Secure Information Flows

Mudhakar Srivatsa, Shane Balfe*, Kenny Patterson*, Pankaj Rohatgi
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

*Royal Holloway
University of London

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Trust Management for Secure Information Flows

ABSTRACT

In both the commercial and defence sectors a compelling need is emerging for the rapid, yet secure, dissemination of information across traditional organisational boundaries. In this paper we present a novel trust management paradigm for securing inter-organisational information flows that aims to address the threat of information leakage. Our trust management system is built around an economic model and a trust-based encryption/decryption primitive wherein: (i) entities purchase a key from a Trust Authority (TA) which is bound to a voluntarily reported trust score r, (ii) information flows are encrypted such that a flow tagged with a recipient trust score R can be decrypted by the recipient only if it possesses the key corresponding to a voluntarily reported score $r \leq R$, (iii) the economic model (the price of keys) is set such that a dishonest entity wishing to maximise information leakage is incentivised to report an honest trust score r to the TA. This paper makes two important contributions. First, we quantify fundamental tradeoffs on information flow rate, information leakage rate and error in estimating recipient trust score R. Second, we present a suite of key encryption algorithms that realise our trust-based encryption/decryption primitive and identify computation and communication tradeoffs between them.

1. INTRODUCTION

Large corporations are slowly being transformed from monolithic, vertically integrated entities, into globally disaggregated value networks, where each member focuses on its core competencies and relies on partners and suppliers to develop and deliver goods and service. The ability of multiple partners to come together, share sensitive business information and coordinate activities to rapidly respond to business opportunities, is becoming a key driver for success.

The defence sector too, has similar, dynamic information sharing needs. The decentralised, dynamic and distributed threat of global terrorism has created a need for information sharing between intelligence agencies of different countries and between multiple security and law-enforcement agencies within a country. Furthermore, traditional wars between armies of nation-states are being replaced by highly dynamic missions where teams of soldiers, strategists, logisticians, and support staff, drawn from a coalition of military organisations as well as local (military and civilian) authorities, fight against elusive enemies that easily blend into the civilian population [40]. Securely disseminating mission critical tactical intelligence to the pertinent people in a timely manner will be a critical factor in a mission's success.

While, information sharing across traditional organisational boundaries is becoming a necessity, it is important to mitigate the risk of unauthorised information disclosure. Such leakage can create the risk of legal liability, financial loss, tarnished reputation, or in some environments, a loss of life. Managing the risk of sensitive information leakage even within a single organisation remains a difficult task. Doing so across multiple organisations with potentially conflicting objectives, loyalties or ideologies represents a grand challenge.

Within a single organisation, it is possible to allow sharing of information while managing the risk of information disclosure by appropriately labelling (or classifying) information with its secrecy characteristics and performing an in-depth security assessment of its, systems and users to create controls necessary to protect information, commensurate with its label. Such a security/risk assessment will typically comprise a number of stakeholders and be carried out in a number of stages, including: system characterisation, threat and vulnerability identification, control analysis, likelihood determination and impact analysis [43]. Subsequently, policies, can be put in place that will permit information to be shared within different parts of the organisations, provided that the recipient has necessary controls in place to protect the information. However, this approach is viable only for enabling routine information sharing scenarios in a relatively static organisation. This approach may not be viable for information sharing across organisations, even in a static setting, as one organisation will typically not permit another to perform a security assessment of its internal systems, controls and people. In dynamic organisations, where systems and processes evolve rapidly and there are transient needs for sharing tactical, time-sensitive information across organisational boundaries, a new method of securing information flows is required.

In this paper we present a novel trust management paradigm for securing both intra- and inter-organisational information flows against the threat of information disclosure. We propose a novel approach for assessing risk in terms of trustworthiness, improve risk estimation by involving estimates of trust, provide a natural mechanism to handle risk transfer across organisations and provide an economic mechanism that forces rational entities towards being honest.

Our approach is based on the following key idea that links economics, key encryption and trust management systems. Entities in our system purchase keys from a Trust Authority (TA) that are tagged with a certain level of entity trust. For example, entity A can purchase a key K with trust level r $(0 \leq r \leq 1)$. The higher the trust level on the key, the cheaper it is to purchase. When B sends information to A, it encrypts it with respect to its own trust estimate R for A using a novel *trust-based encryption scheme* to form a ciphertext. Our encryption scheme has the property that A is able to decrypt such a ciphertext if and only if $r \leq R$. Therefore, A could purchase a cheap, high trust key from the TA, but then would not be able to decrypt anything from B if B does not believe that A is so trustworthy. With this basic approach, we derive a pricing function for keys wherein the optimal policy for a dishonest entity is to report a honest trust score to the TA. In addition, we show how assessing risk in terms of trustworthiness presents a means of throttling information flow where low inter-organisational trust exists. The more information an entity leaks, the less information will be sent to that entity over time. We also analyse and provide guidance on how to safely handle the impact of errors in the monitoring process. Our trust management system is resilient to attacks such as shilling and collusion and may be of independent interest in other settings.

We present a set of options for realising our trust-based encryption primitive, allowing different trade-offs between computation and communication. We give a simple, symmetric key based approach which has very low computational complexity but high communication costs and which requires an on-line TA. This scheme would be suitable for most commercial settings. We then provide a more sophisticated asymmetric approach that combines techniques from ID-based and attribute-based encryption. This approach has lower communication complexity and allows the TA to be off-line for all purposes except the distribution of private keys. This makes it more suitable for use in military scenarios, where minimising communication costs in battery-powered mobile ad hoc networks is of paramount importance.

Our approach relies on two assumptions, firstly, that each organisation has monitoring, or other systems, in place to estimate the probability of information leakage from its partners and secondly that partners benefit from maintaining long term relationships and trust with each other and so would be at worst trying to maximise information leakage while maintaining sufficient trust. The first assumption could be realised if an entity adds watermarks to the information it shares with others or dissemintates decoy information and uses these to detect potential leakage from a partner. A related strategy would be to monitor "underground" activity around obtaining sensitive information. Clearly this approach may not provide accurate results, and our model will assume an additive error in the detection probability. However, if a partner is leaking a significant fraction of the information, we believe this is likely to be detected. Our second assumption, is also realistic, since the cooperating entities are likely to be well respected businesses and military organisations that have a need to maintain their reputation. In this setting, we provide the best strategy for setting information flow limits and information pricing schemes that an entity can use to share information with another partner. This strategy maximises the flow while bounding the information risk, taking into account the errors in the entity's monitoring process with respect to this partner.

Organisation: This paper is structured as follows. In section 2 we examine related work. In section 3, we present our trust management proposal and our economic model. In section 4, we describe our trust-based encyption schemes and discuss their computation and communication overheads. In section 5, taking mobile ad hoc netoworks as a concrete example of a highly dynamic organisation, we look at the constraints imposed by these networks and highlight how our proposal can meet them. We also discuss some open issues in our system. We conclude with section 6.

2. RELATED WORK

2.1 Information Flows

There has been significant research on decentralised information labels and assured information sharing within and across multiple organisations [33, 35, 36, 49, 44, 46] in recent years. However, these works primarily focus on the problem of specifying and manipulating the sharing, propagation and downgrading constraints on the data. These works also assume appropriate security controls that manipulate, bind and respect these labels are already in place, for example, via a secure distributed runtime language, or some other form of a secure distributed trusted computing base. Clearly, in practice, for the settings described in the introduction, one partner cannot be sure of either the existence, or the proper usage of, a secure runtime environment of another partner.

Recently, new approaches based on risk estimation and economic mechanisms have been proposed for enabling the sharing of information in dynamic environments [16, 37]. These approaches are based on the idea that the sender dynamically computes an estimate of the risk of information disclosure in providing information to a receiver based on the secrecy of the information to be divulged and the sender's estimate on the trustworthiness of the recipient. The sender then "charges" the receiver for this estimated risk. The recipient, in turn, can decide which type of information is most useful to him and pay only to access that. Entities would either be given a line of risk credit, or adopt a market-based mechanism to "purchase risk" using a pseudocurrency. Under the assumption that the line of risk credit or the risk available for purchase in the market is limited, an entity will be encouraged to be frugal with their risk credits and, consequently, reluctant to spend it unnecessarily. Since all information flows are "charged" for expected losses due to unauthorised disclosure and the amount of risk available is limited, an argument is made that the total information disclosure risk incurred by an organisation is controlled.

While, as a concept, using risk estimation, charging for risk of information flows and limited risk credits is a promising idea for enabling information sharing in dynamic environments, the existing work in this area [16, 37] has gaps in how this concept can be realised to enable cross organisational secure information flows in dynamic environments such as between organisations or partners in a coalition. Firstly, in both [16, 37], while risk is estimated based on the "trustworthiness" of the recipient, the actual formulas or examples use static credentials (e.g., the security clearance or category set) of the recipient, rather than a dynamic trust metric that depends on behaviour. It is our belief that the degree of trust, gradually built over time between cooperating groups from different organisations both in commercial and military settings, is a better predictor of future behaviour of a partner than the static credential. Secondly, the argument that basing charging on risk estimates is sufficient to ensure that overall risks are controlled, while intuitive, is not proven, and there is no analysis of how errors in the risk estimates impact the bounding of the overall risk.

2.2 Incentive Mechanisms

The use of incentive mechanisms as a means of encouraging behavioural conformity in ad hoc groups has been extensively studied in recent years. Thus far, the goals of such work have been to either reward "good" behaviour [2, 17], or punish "bad" behaviour [21, 15, 26, 45].

For example, approaches, such as those found in [11, 50, 10, 34], attempt to encourage good behaviour through incentive mechanisms. In [11, 50], entities exchange tokens as a means of charging/rewarding service provision/usage. Entities which behave correctly and forward packets are rewarded with additional tokens which, in turn, may be spent on forwarding their own packets. Other incentive mechanisms rely on reputation as a means of encouraging entities to behave correctly. Reputation systems, such as [10, 34], aim to encourage good behaviour by maintaining a trust/reputation score for some subset of entities in a network. If the reputation value for an entity drops below a predefined threshold, then that entity is deemed to be misbehaving and packets from that entity may be probabilistic dropped until the entity starts to conform [25].

By contrast punishment mechanisms, such as those found in [26, 45], typically focus on the permanent exclusion of misbehaving entity from the network. Much like reward-based schemes, punishment strategies typically rely on implementing a threshold scheme, where, once a specific (mis)trust value is reached, an entity may instigate a revocation procedure. Our work differs from these approaches as we are not incentivising entities to conform to a prevailing standard of "goodness". Instead we are incentivising rational entities to be honest in self-assessing their trustworthiness which in turn will affect their ability to access sensitive information.

2.3 Range Queries Over Encrypted Data

One of our approaches to building trust-based encryption in the asymmetric setting (called TIBE in section 4) is based on ideas of [41] for constructing encryption schemes supporting range queries over encrypted data, but with significant improvements in efficiency and security. In essence, to build a TIBE scheme, we are only interested in one-dimensional range queries for intervals of the type $[r, 1)$. Thus our TIBE primitive is a special case of the MRQED primitive of [41]. We still use tree-based methods as in [41], but instead of repeatedly using an IBE scheme to encrypt the same plaintext message as in [41], we use the KEM-DEM paradigm and exploit the common plaintext to gain efficiency. Specifically, we are able to use ID-based KEMs from [5] which are actually designed to be efficient when encapsulating the same symmetric key to multiple recipients. Our approach also allows security to be analysed more easily than in [41], since the technicalities of handling security for multiple encryptions of the same message are already taken care of in [5]. We note that [41] makes no mention of this delicate security issue.

Our approach does not seek to hide the trust value R used during encryption from outsiders (since privacy of trust values is not required in our application). This allows a wider variety of cryptographic components to be employed than in [41], since we no do not require anonymous encryption schemes. For example, it allows us to use the constant size ciphertext hierarchical IBE scheme from [7] to obtain a TIBE scheme with compact ciphertexts. This possibility was also noted in [41]. Neither does our approach attempt to hide the trust value R from the intended recipient, a nice feature of further related work on Hidden Vector Encryption [9]. Again, this simplifies our problem.

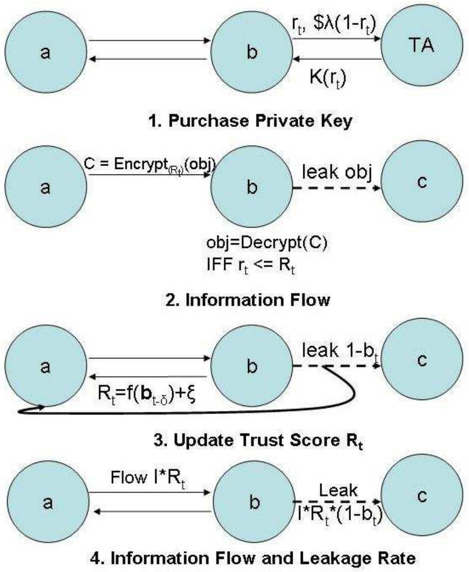
a
b
TA
r_t, $\$\lambda(1-r_t)$
$K(r_t)$
1. Purchase Private Key
a
$C = Encrypt_{(R_t)}(obj)$
b
leak obj
c
obj=Decrypt(C)
IFF $r_t <= R_t$
2. Information Flow
a
b
leak $1-b_t$
c
$R_t=f(\mathbf{b}_{t-\delta})+\xi$
3. Update Trust Score R_t
a
Flow $I*R_t$
b
Leak
$I*R_t*(1-b_t)$
c
4. Information Flow and Leakage Rate

Figure 1: Trust Management Scheme

3. TRUST MANAGEMENT

In this section, we present our trust management system for securing cross-domain information flows. We operate on an information sharing model, wherein a recipient of information is obligated not to divulge received information (in any form) to other participants (henceforth, called entities in this paper). We define an honest entity to be one which obeys its obligation; nonetheless, some information may leak unintentionally from an honest entity. In contrast, a dishonest entity is one which will attempt to maximise the amount of information it leaks.

The goal of our trust management system is to estimate the trustworthiness of an entity. In this context, the trust score for an entity is a measure of the recipient's ability to meet its obligation on shared information. For the sake of simplicity, we quantise information into *objects*. Objects, in turn, are classified into quantised security levels (e.g.: `unclassified, classified, secret, top secret`) based on their value. One could use different trust management models (and parameters) to manage objects at different security levels. For the sake of simplicity, the rest of this paper assumes that we only have one class of objects. The trust score for an entity equals $1-$*the fraction of objects leaked* (intentionally or unintentionally) by that entity. Time is quantised into intervals numbered 0, 1, $\cdots$. The trust score for a entity is updated at the beginning of each time interval.

To detect information leakage, entities will use a monitoring system that estimates the amount of information leaked by a recipient. Our trust management protocol is built around an economic model that forces a rational entity to honestly report its trust score to a TA. An honest entity is compelled to monitor unintentional information leakage; a dishonest entity is compelled to reveal its (intentional) information leakage to the TA, albeit adjusted in favour of any error in the leakage monitoring system. In this section, we first describe our trust management protocol and present a detailed analysis. Our analysis is geared towards identifying a pricing model that compels entities to report honest self-assessment to the TA and quantifying trade-offs between the error in leakage monitoring system and the leakage rate. We show that we can set up the pricing model such that *honesty is the optimal policy even for dishonest entities.*

This distinguishes our trust management scheme from others in the literature that rely on collective opinion to estimate an entity's trust score. For example, approaches such as [28, 47], rely on complex computations that span the entire network to estimate the trust score for an entity. Indeed, some of the biggest drawbacks in relying on collective opinion are *shilling attacks* and *collusion* attacks [47]. In a shilling attack, a group of dishonest entities 'bad mouth' an honest entity, whilst in a collusion attack, the colluding entities attempt to boost one another's trust scores by providing positive feedback to each other. In contrast, our approach compels a dishonest entity to report an honest trust score to the TA if it wishes to engage in an information flow with at least one honest entity in the network (irrespective of the number of colluders). We note that information flow between colluding entities is of no significance to the system. Hence, an entity A interested in sharing information with entity A_i, uses the trust score as voluntarily reported by A_i to the TA, which for a rational A_i would indeed be its honest trust score.

3.1 Protocol

Let $A = \{A_i,\ A_j,\ \ldots\ A_n\}$ denote a set of n entities participating in an organisation. Let TA denote a trusted authority. For the sake of simplicity, we assume there is only one TA in the network. The trust management system operates in rounds. As shown in Figure 1, in each round, the protocol performs four steps: (i) purchase key, (ii) exchange information, (iii) update trust score, and (iv) throttle future information flow rate.

Purchase Key: At the beginning of each round, each entity purchases a key from the TA. In round t, an entity A_i *claims* that its trust score is r_t ($0 \leq r_t \leq 1$) to the TA. A good entity A_j may report an honest self-assessment of itself to the TA. On the other hand, a malicious entity may choose r_t to the best of its interest (as determined by its utility model described later). The TA charges $\lambda(1 - r_t)$ currency units to the entity A_j in exchange for the key associated with trust score r_t ($\lambda > 0$). Entities may use any standard electronic payment mechanism to make this transaction. The quantity λ is determined by a pricing model described in the next section. We note that it is cheaper to buy a key for larger r_t. Hence, a malicious entity attempting to minimise its expense may claim $r_t = 1$, thereby incurring zero cost for the key.

Exchange Information: During round t, two entities engage in information exchange as follows. Let R_t ($0 \leq R_t \leq 1$) denote A_i's trust score for A_j. Entity A_i encrypts all objects from A_i to A_j using a key associated with trust score R_t such that entity A_j can decrypt the objects if and only if $r_t \leq R_t$. In section 4 we describe candidate encryption/decryption algorithms that satisfy this property. Hence, if a dishonest entity A_j were to claim an unfairly high trust score $r_t > R_t$ to the TA, the information flow rate between A_i and A_j (during round t) is zero; on the other hand, if an entity A_j underestimates its trust score $r_t < R_t$, then the entity A_j pays a higher price to purchase the required key from the TA. Intuitively, it appears that the optimal policy for a dishonest A_j would be to set $r_t^* = R_t$. In the next section, we show that this claim may not hold for all pricing models (arbitrary choices of λ) and derive a pricing model whose optimal solution is indeed $r_t^* = R_t$.

Similarly, a dishonest sender A_i may unfairly assign a low trust score $R_t < r_t$ to an honest recipient A_j. In this case, the honest recipient A_j refuses to engage in information exchange with the dishonest sender A_i. We note that R_t is defined pairwise between A_i and A_j. Hence, an honest recipient would be able to engage in information flows with other honest senders. Finally, if both A_i and A_j are honest (or both are dishonest) then this step if trivial.

Update Trust Score: In round t, a malicious entity A_j leaks a fraction ($1 - b_t$, $0 \leq b_t \leq 1$) objects obtained from A_i. Entity A_i monitors and infers information leakage from entity A_j, using either audit logs, interrogating other entities or via some domain-specific leakage detection mechanism. For instance, a monitoring system in a web services application has to measure Quality of Service (QoS) violations, a monitoring system in a tier-1/tier-2 provider network has to measure IP Service Level Agreement (SLA) violations.

In this paper, we do not examine concrete monitoring techniques. Instead, we model a monitoring system as follows: (i) The monitoring system has a finite lag δ between actual information leakage to that of leakage detection; at time t, b_0, …, $b_{t-\delta}$ is observable, while $b_{t-\delta+1}$, …, b_t is unobservable by the monitoring system. (ii) The monitoring system may not be able to accurately observe b_t; the monitoring system outputs $o_t = b_t - err_t$, where err_t is a random variable that represents an additive error in estimating b_t ($|err_t| \leq 1$).

Entity A_i computes its trust score for A_j as $R_t = f(\tilde{o}_{t-\delta})$, where $\tilde{o}_{t-\delta}$ denotes the historical values $\{o_0, o_1, \ldots, o_{t-\delta}\}$. In this paper, we focus only on linear functions f such as moving average, exponentially weighted moving average, etc. Under the linearity assumption, one can rewrite $R_t = f(\tilde{b}_{t-\delta}) - \xi_t$ ($|\xi_t| \leq 1$). In the following sections, we compare candidate functions f and establish bounds on the information leakage rate as a function of the error term ξ_t.

Throttle Information Flow Rate: In round t, an entity A_i throttles its information flow to entity A_j as IR_t ($I > 0$). Hence, if A_i's trust score for A_j is high, then the information flow rate from A_i to A_j is high. Consequently, information leakage by A_j in round t is $IR_t(1 - b_t)$ if $r_t \leq R_t$ and zero otherwise. The parameter I is used by an honest entity to control information flow rate and maximum information leakage rate (as shown in the next section).

Utility Model: Based on the protocol described above, we summarise a simple utility model for a dishonest entity as shown in Equation 1. We assume that a dishonest entity obtains one currency unit per object leaked. Hence, at time t, the utility from information leakage is $IR_t(1 - b_t)$ if $r_t \leq R_t$; zero otherwise. Note that if $r_t > R_t$ the recipient cannot decrypt the received objects; we assume that the encryption mechanisms are sufficiently strong such that the utility from leaking encrypted objects is zero. The cost of purchasing a key from the TA corresponding to a trust score r_t is $\lambda(1 - r_t)$ currency units (one can also think of λ as the ratio of the cost of a key to the profit obtained by leaking one object). Hence, U_t (in Equation 1) denotes the net utility (in currency units) for a dishonest entity at time t. It follows from the structure of U_t that a dishonest node has a myopic incentive to lie about

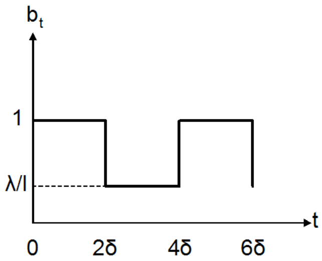

Figure 2: Optimal Policy for Dishonest entity for a Simple Trust Function $-$ $f(\tilde{b}_{t-\delta}) = b_{t-\delta}$

r_t to the TA. Indeed, the monitoring system has no visibility into the last $\delta > 0$ time units, hence a dishonest node can get away with false claims for at least δ time units. Fortunately, in the context of secure information flows, relationships between two entities are long lasting. Hence, we are interested in long term ($\gg \delta$) expected utility for a dishonest entity denoted by $\hat{U}$. The decision variables for a dishonest entity are $\{b_t,\ r_t\}$ for all t; while the decision variables in control of the system are $\{I,\ \lambda,\ f(\cdot)\}$. The error ξ_t is an intrinsic property of the monitoring system and is assumed to be independent of b_t. We assume that the system parameters are public. We use $\tilde{\beta}_t$ to denote historical values $\{\beta_0, \ldots, \beta_t\}$ and $\hat{\beta}$ to denote a long term average of β_t (as defined in Equation 1).

$$\begin{aligned}
&\text{Max} \quad \hat{U} \text{ subject to} \\
U_t \quad &= \quad \begin{cases} IR_t(1-b_t) - \lambda(1-r_t) & \text{if } r_t \le R_t \\ -\lambda(1-r_t) & \text{if } r_t > R_t \end{cases} \\
R_t \quad &= \quad f(\tilde{b}_{t-\delta}) - \xi_t \\
\hat{U} \quad &= \quad \lim_{T\to\infty} \frac{\sum_{t=0}^{T-1} U_t}{T} \qquad (1)
\end{aligned}$$

3.2 Analysis

In this section, we present an analysis of our trust management scheme for a two entity scenario with entity A_i as the information source and entity A_j as the information sink. We develop a pricing model, namely, choice of I, λ for candidate functions f, such that all entities report an honest self-assessment to the TA. We say that r_t, a self-assessment of an entity A_j, is honest if $r_t = R_t$, the trust score assigned by an honest entity A_i to A_j. We assume that the dishonest entities are rational, that is, they attempt to maximise the objective in Equation 1. Hence, a rational dishonest entity is compelled to reveal an honest trust score if $r^*_t = R_t$, namely, the optimal choice of the decision variable r_t equals R_t.

Analysing U_t with respect to the decision variable r_t, it is easy to see the following:

$$U^*_t \quad = \quad \begin{cases} 0 & \text{if } r_t > R_t \text{ at } r^*_t = 1 \\ IR_t(1-b_t) - \lambda(1-R_t) & \text{if } r_t \le R_t \text{ at } r^*_t = R_t \end{cases} \qquad (2)$$

Let $Y_t = IR_t(1-b_t) - \lambda(1-R_t)$. In order to ensure that a dishonest entity reports an honest self-assessment, we require $r^*_t = R_t$. Hence, we require $Y_t > 0$, $\forall t$. If $Y_t \le 0$, for some t, then a dishonest entity would choose $r^*_t = 1 > R_t$ in order to maximise its utility $\hat{U}$.

We examine $\hat{Y}$ for a simple candidate trust function $f(\tilde{b}_{t-\delta})$ $= b_{t-\delta}$ (that does not work). We can maximise $\hat{Y}$ with respect to the decision variable b_t, by setting $\frac{\partial \hat{Y}}{\partial b_t} = 0$ for all t. Note that b_t (for a given t) occurs exactly twice in $\hat{Y}$, namely, $Y_t = Ib_{t-\delta}(1-b_t) - \lambda(1-b_{t-\delta})$ and $Y_{t+\delta} = Ib_t(1-b_{t+\delta}) - \lambda(1-b_t)$. Setting $\frac{\partial \hat{Y}}{\partial b_t} = 0$ gives us a system of linear equations shown in Equation 3.

$$b_{t-\delta} + b_{t+\delta} = \frac{I+\lambda}{I}, \forall t \qquad (3)$$

We note that if $\lambda > I$ then the system of linear equations has no solutions since it would require $b_{t-\delta} + b_{t+\delta} = \frac{I+\lambda}{I} > 2$. By pigeon hole principle, either $b_{t-\delta}$ or $b_{t+\delta}$ should be greater than one, and is thus infeasible. Assuming $\lambda \le I$, equation 3 indicates that the optimal policy for a dishonest entity is periodic with periodicity equal to 4δ, that is, $b_t = b_{t+4\delta}$ for all t. The dishonest entity can choose $b_i = x_i$ for any $0 \le x_i \le 1$ for $0 \le i \le 2\delta - 1$; and $b_i = \frac{I+\lambda}{I} - x_i$ for $2\delta \le i \le 4\delta - 1$. Figure 2 shows an oscillatory policy which happens to be one of the optimal policies for a dishonest entity obtained by setting $x_i = 1$ for all $0 \le i \le 2\delta - 1$.

Setting $b_{t-\delta} + b_{t+\delta} = \frac{I+\lambda}{I}$, we observe that $\hat{b_t} = \frac{I+\lambda}{2I}$. Also, $b_{t-\delta} + b_{t+\delta} = \frac{I+\lambda}{I}$ implies $b_{t-\delta}b_t + b_t b_{t+\delta} = \frac{I+\lambda}{I} b_t$. Hence, $\widehat{b_{t-\delta}b_t} = \frac{I+\lambda}{2I}\hat{b_t} = (\frac{I+\lambda}{2I})^2$. Piecing together all the above we find that $\hat{Y} = \frac{(I-\lambda)^2}{4I} \ge 0$. However, we observe from Figure 2 that there may exist t such that $Y_t \le 0$: $R_t = \frac{\lambda}{I}$, $b_t = 1 \Rightarrow Y_t = -\lambda(1 - \frac{\lambda}{I}) \le 0$ for $\lambda \le I$. Hence, a dishonest entity *will* report an incorrect self-assessment (namely, $r^*_t = 1$) when: R_t $= \frac{\lambda}{I} \le 1$. We note that ensuring the long term average $\hat{Y} >$ 0 does not suffice; we require $Y_t > 0$, $\forall t$.

One can fix this problem using a candidate trust function $f(\tilde{b}_{t-\delta}) = \alpha b_{t-\delta} + (1-\alpha) b_{t-\delta-1}$ for some $0 < \alpha < 1$. Using a similar analysis and setting $\frac{\partial \hat{Y}}{\partial b_t} = 0$ for all t, we obtain the following set of linear equations:

$$\alpha(b_{t-\delta} + b_{t+\delta}) + (1-\alpha)(b_{t-\delta-1} + b_{t+\delta+1}) = \frac{I+\lambda}{I}, \forall t \qquad (4)$$

If α is a public known constant parameter one can show that we run into the same problem described above, namely, although $\hat{Y} \ge 0$, there may exist t such that $Y_t < 0$. In fact, a thorough analysis reveals that using $f(\tilde{b}_{t-\delta}) = \sum_{y=0}^{t-\delta} \alpha(1-\alpha)^y b_{t-\delta-y}$, yields identical results for $\hat{Y} = \frac{(I-\lambda)^2}{4I}$, while $\exists t$ such that $Y_t < 0$ irrespective of the choice of α.

The key trick to solve this problem is to not fix the parameter α; instead the entity A_i chooses α uniformly and randomly in the range $(0, 1)$ when evaluating the trust function $f(\cdot)$. Hence, a dishonest entity is forced to optimise $\hat{Y}$ over all possible values of α. Equivalently, a dishonest entity has to solve the set of linear equations in Equation 4 independent of the choice of α. This is accomplished by rewriting Equation 4 as follows:

$$\alpha(b_{t-\delta} + b_{t+\delta} - b_{t-\delta-1} - b_{t+\delta+1}) + (b_{t-\delta-1} + b_{t+\delta+1}) = \frac{I+\lambda}{I}$$

$\Rightarrow$

$$\begin{aligned}
b_{t-\delta} + b_{t+\delta} - b_{t-\delta-1} - b_{t+\delta+1} \quad &= \quad 0 \\
b_{t-\delta-1} + b_{t+\delta+1} \quad &= \quad \frac{I+\lambda}{I} \qquad (5)
\end{aligned}$$

One can show that the system of equations in 5 has a unique solution $b^*_t = \frac{I+\lambda}{2I}$. Substituting b^*_t in Equation 2 we get Y^*_t $= \frac{(I-\lambda)^2}{4I}$. Hence,

$$b^*_t \quad = \quad \begin{cases} \frac{I+\lambda}{2I} & \text{if } \lambda \le I \\ 1 & \text{if } \lambda > I \end{cases}$$

$$U_t^* = \begin{cases} 0 & \text{if } r_t > R_t \text{ at } r_t^* = 1 \\ \frac{(I-\lambda)^2}{4I} & \text{if } r_t \le R_t \text{ at } r_t^* = R_t \end{cases} \quad (6)$$

Indeed, if we choose the pricing model such that $\lambda < I$, $r_t^* = R_t$, that is, a rational dishonest entity will report an honest evaluation to the TA. One can show that using a candidate trust function $f(\tilde{b}_{t-\delta}) = \sum_{y=0}^{t-\delta} \alpha(1-\alpha)^y b_{t-\delta-y}$, yields identical results for U_t^*. This motivates us to hypothesize that leakage measurements over the last two time windows ($t-\delta$ and $t-\delta-1$) is necessary and sufficient for the trust management system. We also hypothesize that a finite lag δ does not affect long term efficacy of the trust management system (some short term efficacy issues are discussed towards the end of this section).

It appears from Equation 6 that by setting λ arbitrarily close (but not equal) to I, one can reduce the utility for a dishonest entity to nearly zero, while forcing the dishonest entity to report an honest self-assessment to the TA. However, this is not true in practice since the leakage detection mechanism and the trust management system is never 100% precise. In the rest of this section, we show that accounting non-zero error term ξ_t in R_t, it may not be feasible to set λ arbitrarily close to I.

In the rest of section, we use $f(\tilde{b}_{t-\delta}) = \alpha b_{t-\delta} + (1-\alpha) b_{t-\delta-1}$ (where α is chosen randomly between (0, 1) during the computation of $f(\cdot)$) and $R_t = f(\tilde{b}_{t-\delta}) - \xi_t$. Using the analysis described above, one can show that $b_t^* = \frac{I(1+\hat{\xi})+\lambda}{2I}$. We note that $\hat{\xi}$ denotes the average error in the trust score estimate by A_i. The expected value of U_t^* is given by the following equation:

$$b_t^* = \begin{cases} \frac{I(1+\hat{\xi})+\lambda}{2I} & \text{if } \lambda \le I(1-\hat{\xi}) \\ 1 & \text{if } \lambda > I(1-\hat{\xi}) \end{cases}$$

$$E(U_t^*) = \begin{cases} 0 & \text{if } r_t > R_t \text{ at } r_t^* = 1 \\ \frac{I^2(1-\hat{\xi})^2+\lambda^2-2\lambda I(1+\hat{\xi})}{4I} & \text{if } r_t \le R_t \text{ at } r_t^* = R_t \end{cases} \quad (7)$$

In order to force a dishonest entity to report an honest self-assessment we require $r_t^* = R_t$, that is, $\frac{I^2(1-\hat{\xi})^2+\lambda^2-2\lambda I(1+\hat{\xi})}{4I} > 0$. This leads us to the following bound on $\hat{\xi}$:

$$\hat{\xi} < \frac{(\sqrt{I}-\sqrt{\lambda})^2}{I} \quad (8)$$

Now, it is easy to see that setting λ arbitrarily close to I may not be feasible in practice since it would require $\hat{\xi} < 0$, that is, the error in trust management system has to be smaller than zero. On the other hand, if λ is set to zero, we get $\hat{\xi} < 1$, that is, there are no constrains on the efficacy of the monitoring system. However, we observe that $\frac{\partial U_t^*}{\partial \lambda} = 2(\lambda - I) < 0$ (for $\lambda < I$), that is, as λ decreases, the utility for an dishonest entity U_t^* increases. Hence, given an estimate on the estimate on the error in the monitoring system ($\hat{\xi} > 0$) the trust management system should pick the largest λ that satisfies Equation 8. The optimal choice for the pricing model parameter λ^* as given by Equation 9. The corresponding optimal strategy by a dishonest entity is also shown below.

$$\lambda^* = \begin{cases} I\left(1-\sqrt{\hat{\xi}}\right)^2 & \text{if } \hat{\xi} > 0 \\ I & \text{if } \hat{\xi} < 0 \\ I - \epsilon & \text{if } \hat{\xi} = 0 \text{ for some small } \epsilon > 0 \end{cases} \quad (9)$$

$$b_t^* = \begin{cases} 1+\hat{\xi}-\sqrt{\hat{\xi}} & \text{if } \hat{\xi} > 0 \\ 1+\frac{\hat{\xi}}{2} & \text{if } \hat{\xi} < 0 \\ 1-\frac{\epsilon}{2I} & \text{if } \hat{\xi} = 0 \text{ for some small } \epsilon > 0 \end{cases}$$

If $\hat{\xi} \le 0$, the system parameter λ can be set arbitrarily close to I, thereby minimising the utility for an dishonest entity, while forcing it to report an honest self-assessment to the TA. Even if $\hat{\xi} < 0$ and λ set to I, $\hat{U} = \frac{I}{4}\hat{\xi}(\hat{\xi}-4)$. Observe that as $\hat{\xi}$ varies from 0 to -1, $\hat{U}$ monotonically increases. Indeed, $\hat{U}$ can be zero if and only if $\hat{\xi} = 0$.

One can also compute the good put $G_t = IR_t b_t$ (IR_t is the information flow rate and b_t is the fraction that is not leaked) and leakage rate $L_t = IR_t(1-b_t)$ as:

$$\hat{G}^* = \frac{(I+\lambda^*)^2-(I\hat{\xi})^2}{4I}$$
$$\hat{L}^* = \frac{I^2-(I\hat{\xi}+\lambda^*)^2}{4I} \quad (10)$$

We note that the optimal good put $\hat{G}^*$ decreases irrespective of $\hat{\xi}$ is +ve or −ve. While it may appear from Equation 10 that $\hat{L}^*$ could be −ve, using the optimal setting for λ from Equation 9, it is easy to see that $\hat{L}^* \ge 0$ (note: $\left(1-\sqrt{\hat{\xi}}\right)^2$ $\le 1-\hat{\xi}$ for $0 \le \hat{\xi} \le 1$). Using the optimal setting of λ from Equation 9, it is easy to see that:

$$\hat{L}^* = \begin{cases} \hat{G}^*\sqrt{\hat{\xi}} & \text{if } \hat{\xi} > 0 \\ \frac{\hat{G}^*}{1-\frac{2}{\hat{\xi}}} & \text{if } \hat{\xi} < 0 \\ \frac{\epsilon}{2} & \text{if } \hat{\xi} = 0 \text{ for some small } \epsilon > 0 \end{cases} \quad (11)$$

This indicates that if $\hat{\xi} \ne 0$, then it is impossible to increase the good put without increasing the leakage rate. We also observe a counter-intuitive result: it is better for the trust management system to overestimate R_t rather than conservatively underestimate it. Note that, $R_t = f(\tilde{b}_{t-\delta}) - \xi_t$ and $\sqrt{\hat{\xi}}$ $> \frac{1}{1+\frac{2}{|\hat{\xi}|}}$ for all $0 \le \hat{\xi} \le 1$. For example, 10% underestimation ($\hat{\xi}$=0.1) results in $\hat{L}^* = 0.32\hat{G}^*$, while 10% overestimation ($\hat{\xi}$=−0.1) results in $\hat{L}^* = 0.05\hat{G}^*$. The key intuition here lies in the choice of λ^* (see Equation 9). If we overestimate (that is, $\hat{\xi} < 0$), then λ can be set close to I while incentivising a rational entity to be honest to the TA; on the other hand, if $\hat{\xi} > 0$, then λ has to be significantly smaller than I in order to incentivise a rational entity to be honest to the TA.

We also note that the trust management scheme promotes an honest entity to monitor and account for unintended information leakage (if any) into r_t. We also recognise that a dishonest entity may deviate from its optimal policy to gain short term utility (though it stands to loose on a longer run). For instance, a dishonest entity A_i may report $r_t = 1$; entities that have no previous interacted with A_i will obtain $R_t = r_t$ = 1 from the TA. Hence, the dishonest entity A_i can engage in an information flow with A_j and subsequently leak the information, thereby achieving short term utility gain. However, soon R_t, as computed by A_j, would drop; if A_i continues to report $r_t = 1$, then its long run utility from A_j would be zero. The duration of short term utility depends on the lag term δ and the error term ξ. A detailed transient analysis of the utility function is outside the scope of this paper.

4. TRUST-BASED ENCRYPTION

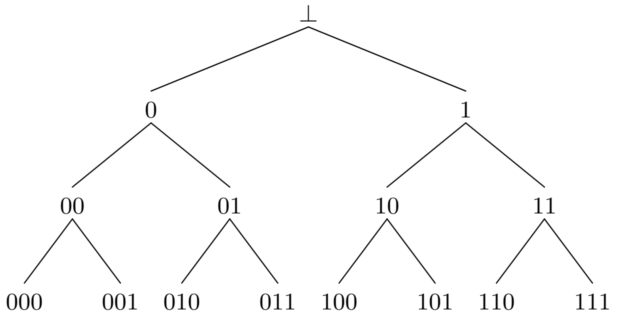

Figure 3: Binary Tree: $\mathcal{S}_{\frac{1}{8}} = \{001,\ 01,\ 1\}$, $\mathcal{R}_{\frac{1}{4}} = \{\perp,\ 0,\ 01,\ 010\}$, $S_{\frac{1}{8}} \cap R_{\frac{1}{4}} = \{01\}$

In this section, we present a suite of cryptographic approaches which realise the encryption functionality discussed in section 3.

The main property we require is that an encrypting entity A_i should be able to specify a trust rating R when encrypting information for an entity A, in such a way that A_j can decrypt only if it is in possession of a secret key whose trust rating r satisfies $r \leq R$. In addition, we wish the keys to depend on temporal information and identities, so that keys for one round are useless in the next round, and so that keys are tied to identities (thus providing suitable key separation).

We begin by sketching a simple approach based only on symmetric key cryptography. This approach has the benefit of being computationally lightweight, but demands that an encrypting entity A_i contact the TA to obtain a suitable key before performing the encryption to A_j; A_j receives information allowing it to compute the same key at the beginning of each round. This means that the solution requires additional communication between A_i and the TA and an on-line TA. It is therefore essentially a TTP-assisted key distribution protocol, but with the added functionality that A_i can specify a trust requirement which A_j needs to meet in order to be able to decrypt.

We then present a more sophisticated public key approach, which combines identity-based encryption (IBE) with techniques from attribute-based encryption to obtain a low interaction solution. Here, A_i needs only know A_j's identity, the round number t and some global system parameters in order to encrypt for A_j with a specified trust rating R; A_j can only decrypt if it has obtained a private key for trust rating $r \leq R$ for round t from the TA. In contrast to the symmetric approach, A_i need not interact with the TA in order to encrypt to A_j. The solutions here are obtained by specialising and stream-lining tree-based techniques for range queries over encrypted data from [41]. We use identity-based key encapsulation techniques designed for multiple recipients from [5], as well as Hierarchical IBE (HIBE), to explore different trade-offs between computation, communication and storage.

Because of space restrictions, we focus here on functionality and on sketching security properties of our schemes. Formal security analysis will follow in a full version.

4.1 Trees and Paths

Both of our approaches make use of binary trees of the type shown in Figure 3. The binary tree of depth d has a root labeled $\perp$ (representing the string of length 0), a left-child at node s labeled $s0$ and right-child labeled $s1$. Thus, the nodes are labeled by binary strings of length (at most) d and the leaves are labeled from left-to-right by d-bit strings, beginning $0\ldots0$ and ending $1\ldots1$. We associate with each binary string $b_0\ldots b_{l-1}$ of length $l \leq d$ (and hence with nodes in the tree) the real number $\sum_{i=0}^{l-1} b_i 2^{-(i+1)}$ in the interval $[0, 1)$.

We assume throughout that each trust rating $r \in [0, 1)$ can be represented as a rational number of the form $r = a/2^k$. Here k is a system parameter representing the granularity of the trust ratings. With each such value r, we associate a set of nodes $\mathcal{S}_r$ in the binary tree of depth k in the following way: corresponding to r is a leaf labeled with the binary string $r_0 \ldots r_{k-1}$ where $r = \sum_{i=0}^{k-1} r_i 2^{-(i+1)}$. We define $\mathcal{S}_r$ to be the set of nodes obtained as the roots of a minimal set of subtrees that exactly cover the leaves with labels $r_0 \ldots r_{k-1}$ up to $11\ldots1$ (in other words, which cover the leaves representing real numbers in the range $[r, 1)$). Some examples should clarify this description. Consider $k = 3$. If $r = 0$, then we need subtrees that cover *all* nodes in the tree, and we can clearly take $\mathcal{S}_0 = \{\perp\}$. If $r = 1/8$, then we need subtrees covering all nodes 001 to 111, representing the interval $[1/8, 1)$. The subtrees rooted at 001, 01 and 1 are a minimal set, and we take $\mathcal{S}_{1/8} = \{001, 01, 1\}$. In general, it is easy to see that the minimal set of rooted subtrees is unique and of size at most k; moreover, it is easy to compute the labels of the nodes in $\mathcal{S}_r$ from the binary expansion of $a = 2^k \times r$.

Finally, given any rational number $R = b/2^k$ with $r \leq R$, we may write $R = \sum_{i=0}^{k-1} R_i 2^{-(i+1)}$ and construct the following path of nodes $\mathcal{P}_R$ in the tree:

$$\perp, R_0, R_0R_1, \ldots, R_0 \ldots R_{k-1}.$$

This path ends at the leaf corresponding to R. It is not hard to see that the path $\mathcal{P}_R$ intersects the set $\mathcal{S}_r$ at a unique node if and only if $r \leq R$. For example, if $k = 3$, $r = 1/8$ and $R = 1/4$, then we have the path $\perp, 0, 01, 010$ which intersects $\mathcal{S}_{1/8}$ at 01.

4.2 Using Symmetric Key Techniques

We present an approach based on hash trees to generate symmetric keys. Our approach allows an encrypting entity A_i to specify a trust rating R when encrypting information for an entity A_j, in such a way that A_j can decrypt only if it is in possession of a secret key whose trust rating r satisfies $r \leq R$.

The TA maintains a master secret key *msk*, and derives all keys from this key using a one-way hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^l$ (where l is some fixed value, say 128). Each pair $(\mathtt{id}_A, t)$, denoting a recipient identifier and a round number, is associated with a distinct tree $\mathcal{T}_{\mathtt{id}_{A,t}}$ of depth k, and each node s of such a tree is assigned a key $K(s)$ by the TA. The root of this tree is assigned the key $K(\perp) = H(msk||\mathtt{id}_A||\mathtt{id}_B||t)$, while we have:

$$\begin{aligned} K(s \parallel 0) &= H(K(s) \parallel 0) \\ K(s \parallel 1) &= H(K(s) \parallel 1) \end{aligned}$$

for every string s.

Entity A_j with trust rating r is given the set of (at most k) keys $\{K(s) : s \in \mathcal{S}_r\}$ by the TA at the beginning of round t. This communication of keys needs to take place over a secure channel between A_j and the TA; we assume these entities maintain a long-term key for this purpose. Entity A_i who wishes to communicate with A_j during round t selects a trust rating $R = \sum_{i=0}^{k-1} R_i 2^{-(i+1)}$ and is given the key $K_{R,N} = H(K(R_0 \ldots R_{k-1})||N)$ by the TA, also over a secure channel. Here N is a random l-bit nonce string which is selected by the

TA and given to A_i along with the key. A_i uses this key for encrypting data intended for A_j during time period t. It is easy to see that, since the path $\mathcal{P}_R$ intersects the set $\mathcal{S}_r$ if and only if $r \leq R$, then A_j can derive the key $K_{R,N}$ for itself from N and its private key set using at most $k+l$ hash operations, provided $r \leq R$. On the other hand, if $r > R$, then it is not possible for A_j to compute the required key from its key set without breaking the one-wayness of H.

The parameter l is chosen so as to provide a form of key separation between different entities A_i: the nonce N of length l ensures that different parties who wish to communicate with A_j in round t receive different keys (with high probability) and cannot pool their keys to obtain an advantage. At the same time, A_j needs only one key set (distributed at the start of the round) in order to receive communications from all entities. An alternative approach to providing key separation would be to use a separate key hierarchy for each possible pair of communicating entities (A, B) at the start of each round. This would involve transporting greater amounts of key information at the start of each round.

Overall, the symmetric key solution presented here is extremely lightweight computationally, requiring only simple hashing operations on the part of the TA and decrypting parties A_j. But it requires the TA to be on-line for distributing keying information to entity A_i, as well as the distribution of fresh keys to entity A_j at the start of each round.

4.3 Using Identity-based Techniques

Next we explain how to use identity-based encryption techniques to construct cryptographic schemes that are suitable for environments where low interaction between communicating nodes and between nodes and the TA is needed.

In this setting, we define a trust-and-identity-based encryption (TIBE) scheme in terms of four algorithms, Setup, KeyDer, Encrypt, Decrypt.

Setup takes as input a security parameter 1^ℓ and outputs system parameters params, which includes specifications of message, ciphertext, identity and private key spaces, as well as a master secret *msk* and a maximum granularity parameter k. KeyDer takes as input an arbitrary identity string id and a trust value $r \in [0, 1)$, along with *msk*, and outputs a corresponding private key $d_{\mathtt{id},r}$ for that identity. As usual, we assume that r is a rational number of the form $a/2^k$ in the range $[0, 1)$. Setup and KeyDer are normally run by the TA in our scheme.

The input to Encrypt is a pair $(\mathtt{id}, R)$, params, and a message M, and its output is a ciphertext C. Decrypt is the corresponding decryption algorithm; its input is a ciphertext C and a private key $d_{\mathtt{id},r}$ and its output is either a message M or a failure symbol $\perp$. We have an obvious consistency requirement: if C is obtained by encrypting M for id with rating R, then Decrypt outputs M on input C and $d_{\mathtt{id},r}$, provided $r \leq R$, and $\perp$ otherwise.

Notice that we do not make explicit here the round parameter t. We simply assume that identities are extended to include the round number t; the security notions we develop next ensure that private keys in any given round t are useless in every other round.

Security for a TIBE scheme can be defined by extending the usual security game for IBE [8]. In the IND-CCA setting, we give an adversary access to a decryption oracle, a key extraction oracle (which returns a private key $d_{\mathtt{id},r}$ when given as input $(\mathtt{id}, r)$), and a challenge oracle. The challenge oracle is called once by the adversary, who specifies as input two messages M_0, M_1, and a pair $(\mathtt{id}^*, R^*)$. The response is the encryption C^* of M_b under $(\mathtt{id}^*, R^*)$, for $b \xleftarrow{\$} \{0, 1\}$. The adversary's job is to output a bit b', and is deemed successful if $b' = b$. The adversary is not permitted to make a decryption query on C^* during the game, and is not permitted to make a key extraction query on any pair $(\mathtt{id}^*, r)$ with $r \leq R^*$. Such queries would allow the adversary to trivially win the security game; all other queries are permitted. The adversary's advantage is defined to be $|\Pr(b' = b) - 1/2|$. A TIBE scheme is said to be IND-CCA secure if no polynomial time adversary has non-negligible advantage in this security game (as a function of ℓ).

Weaker notions of security can be defined, e.g. selective-ID, and IND-CPA security. These need not concern us here. We could also extend this definition to ensure that ciphertexts do not reveal the identity id or the trust rating R used in their preparation. This can be formalised in a similar way to the notion of anonymity for IBE, and is related to the notion of hiding of attributes in attribute-based encryption. We leave this as a topic for future work.

It should be clear how a secure TIBE scheme meets our need that only a recipient with a private key for trust rating $r \leq R$ can decrypt a ciphertext prepared using a trust rating R in round t. This application of TIBE will necessitate the provision of a secure channel for distributing fresh private keys to nodes at the start of each round t. We may assume that each node maintains an ID-based key for this purpose. As an alternative, the lightweight key refreshing techniques of [4] can be used.

4.3.1 Construction of TIBE from MR-SK-IBKEM

An ID-based Key Encapsulation Mechanism (KEM) [6] is a generalisation of an ID-based encryption (IBE) scheme. It is defined in terms of 4 algorithms, Setup, KeyDer, Encap, Decap. Setup outputs system parameters params and master secret *msk*, KeyDer takes as input an arbitrary string id and outputs a corresponding private key $d_{\mathtt{id}}$. Encap is an encapsulation algorithm; it's input is a string idand params, and its output is a pair (K, c) where K is a symmetric key from some keyspace and c is an *encapsulation* of that key. Decap takes as input an encapsulation c together with a private key $d_{\mathtt{id}}$ and outputs either a key K or the failure symbol $\perp$. There is an obvious consistency requirement: if (K, c) is output by Encap on input $(\mathtt{id}, \mathtt{params})$, then Decap outputs K on input $(c, d_{\mathtt{id}})$.

An ID-based KEM can be combined with a (symmetric) Data Encapsulation Mechanism (DEM) to produce an IBE scheme in a standard way [6]; if the KEM and DEM satisfy appropriate security notions (IND-CCA and FG-CCA security, respectively), then the resulting IBE scheme will be IND-CCA secure [6].

A multi-recipient, single key, ID-based KEM (MR-SK-IBKEM) as defined in [5] is a generalisation of the ID-based KEM notion that allows encapsulation of the same key K for multiple recipients $\mathtt{id}_1, \ldots, id_m$ in an efficient and secure manner. An MR-SK-IBKEM is also defined in terms of 4 algorithms, Setup, KeyDer, Encap, Decap. Setup outputs params and *msk*, KeyDer takes as input an arbitrary string id and *msk*, and outputs a corresponding private key $d_{\mathtt{id}}$. Encap is an encapsulation algorithm; its input is a list of distinct strings $\mathtt{id}_1, \mathtt{id}_2, \ldots, \mathtt{id}_m$ together with params, and its output is a tuple $(K, c_1, \ldots, c_m)$ where K is a symmetric key from some keyspace and c_i is an *encapsulation* of that key for identity $\mathtt{id}_i$. Decap takes as input an encapsulation c together with a private key $d_{\mathtt{id}}$ and outputs either a key K

or the failure symbol $\perp$. There is again a consistency requirement: if $(K, c_1, \ldots, c_m)$ is output by Encap on input $(\mathtt{id}_1, \ldots, id_m, \mathtt{params})$, then Decap outputs K on input $(c_i, d_{\mathtt{id}_i})$ for each i. The security model for MR-SK-IBKEM is obtained by appropriately modifying the ID-based KEM security model, and a similar composition theorem shows that an MR-SK-IBKEM can be combined with a DEM to get a strongly secure ID-based scheme which encrypts the same message for multiple recipients [5].

We sketch how to build a TIBE scheme from any MR-SK-IBKEM and any DEM. We then discuss efficient instantiations of this construction.

Setup for our TIBE scheme simply replicates Setup of the underlying MR-SK-IBKEM scheme, but also outputs as part of the system parameters the granularity value k. KeyDer operates as follows. It has as input an identity id and a trust rating r. Its output is the set $\mathcal{D}_{\mathtt{id},r}$ of private keys from the MR-SK-IBKEM scheme corresponding to identities $\{\mathtt{id}||s : s \in \mathcal{S}_r\}$. This set contains at most k private keys. (Note that when $r = 0$, $\mathcal{D}_{\mathtt{id},r}$ just contains the private key for identity id.)

Now Encrypt works as follows. Given a message m, an identity id and a trust rating R, we construct the path $\mathcal{P}_R$ containing $k+1$ nodes. We then run the Encap algorithm of the MR-SK-IBKEM scheme on input the set of $k+1$ distinct identities $\{\mathtt{id}||p : p \in \mathcal{P}_R\}$ to obtain an output of the form $(K, c_0, \ldots, c_k)$. We then run the encryption algorithm of the DEM on input key K and message m to obtain a ciphertext C. The final output of Encrypt is the tuple $(c_0, \ldots, c_k, C)$.

Decrypt operates as follows. Because of the way private keys are assigned, an entity with identifier id and trust rating $r \leq R$ has private keys $\mathcal{D}_{\mathtt{id},r}$, and so has a private key corresponding to the intersection of $\mathcal{S}_r$ and $\mathcal{P}_R$. Say this corresponds to node i in the path $\mathcal{P}_R$; then the entity can run Decap of the MR-SK-IBKEM scheme with inputs of this private key and c_i to obtain a key K. The entity then runs the decryption algorithm of the DEM with key K and ciphertext C to obtain M.

Security and Efficiency: The security guarantees of the underlying MR-SK-IBKEM scheme are sufficient to ensure that the resulting TIBE scheme is secure in the model outlined in Section 4.3. The full version will include a proof of the following:

THEOREM 4.1. *Suppose we have an IND-CCA secure MR-SK-IBKEM and an FG-CCA secure DEM. Then the TIBE scheme obtained from these components using our generic construction is IND-CCA secure.*

We note in passing that a similar construction for TIBE can be obtained using a normal IBE scheme or ID-based KEM in place of the MR-SK-IBKEM scheme. However, the usual security notions for IBE would not be strong enough to guarantee security of the TIBE scheme as in the above theorem. This is because those notions do not say anything about what happens when a single message is encrypted to multiple parties. In the more general context of range queries over encrypted data, this represents a flaw in the security reasoning for the IBE-based schemes in [41, Section 4].

The performance of our TIBE construction is determined by that of the underlying MR-SK-IBKEM and the granularity parameter k. Decryption requires one decapsulation computation; encryption runs the Encap algorithm of the MR-SK-IBKEM just once; ciphertext size is determined by that of the MR-SK-IBKEM scheme. The size of a private key set $\mathcal{D}_{\mathtt{id},r}$ can be as large as k private keys in the original MR-SK-IBKEM scheme, often it is less.

A particularly efficient MR-SK-IBKEM using pairings is given in [5, Section 5.2]. The scheme there works in the setting of asymmetric pairings $e : G_1 \times G_2 \rightarrow G_T$ and is based on Smart's OR construction. Security is based on the hardness of a gap Bilinear Diffie-Hellman Problem. Using this MR-SK-IBKEM, we obtain a TIBE scheme in which encryption needs just one pairing computation and $k+3$ scalar multiplications in G_1 (together with symmetric operations), and decryption is dominated by the cost of two pairing computations. Ciphertexts consist of $k+3$ elements of G_1 together with the DEM part of the ciphertext, and private keys contain at most k elements of G_2 (and are often much smaller). This is certainly efficient enough for use in practice. For example, instantiating the TIBE scheme using a supersingular curve of embedding degree 6 defined over $\mathbb{F}_{3^{163}}$, we get roughly 80 bits of security, while elements of G_1 and G_2 can each be represented using 260 bits and each pairing computation can be done in 57ms on a 2.4GHz Pentium-4 platform [38]. For $k = 4$, for example, the ciphertext overhead is 1820 bits and private keys are 1040 bits.

An alternative approach in [22, Chapter 5] indicates how to build an efficient MR-SK-IBKEM from any IND-CPA secure IBE scheme that is *weakly reproducible*. The BasicIdent scheme of [8] is a suitable candidate. The result for us is a TIBE scheme that requires more computation in encryption but that has slightly reduced ciphertext overhead.

Dynamic Granularity: Our TIBE construction allows the granularity k to be selected dynamically by encrypting parties to save bandwidth. If an entity wishes to work at granularity $k' < k$, that is with trust ratings R of the form $b/2^{k'}$, then he can do so and needs only prepare the shorter ciphertext $(c_0, \ldots, c_{k'}, C)$; Decrypt will work just as before. Here, the encrypting party effectively encrypts for identities along a path to an internal node in the tree rather than a leaf. This internal node still defines an interval of the form $[R, 1)$.

Similarly, an entity with constrained storage can request a private key set for a trust rating r having granularity $k' < k$. Now the private key set $\mathcal{D}_{\mathtt{id},r}$ will contain k' private keys, and the entity can still decrypt all ciphertexts encrypted with a trust rating R satisfying $r \leq R$.

4.3.2 Construction of TIBE from HIBE

Our construction of a TIBE scheme from an MR-SK-IBKEM involves, for all instantiations of which we are aware, a degree of ciphertext expansion (in all cases, the ciphertexts grow linearly with k, the granularity parameter). Here we sketch an alternative approach based on HIBE [23] which can avoid this expansion. In HIBE, we have a hierarchy of TAs, with a Root TA located at level 0, and each TA issuing private keys to entities at the level below it in the hierarchy. An entity at level l now has an identity of the form $(\mathtt{id}_1, \mathtt{id}_2, \ldots, \mathtt{id}_l)$, where $(\mathtt{id}_1, \mathtt{id}_2, \ldots, \mathtt{id}_{l-1})$ is the identity of its immediate ancestor, etc.

We use HIBE supporting $k+1$ levels to construct TIBE with granularity k as follows. The Root TA in HIBE plays the role of the TIBE TA, and entities with identifiers id play the role of TAs at level 1. Entities at levels 2 down to $k+1$ below each entity are identified by binary strings and form a binary tree as in Section 4.1. Now an entity with trust rating r is given the set $\mathcal{D}_{\mathtt{id},r}$ of private keys from the HIBE scheme corresponding to the set of identities $\{(\mathtt{id}, s_1, s_2, \ldots, s_l) : s_1 s_1 \ldots s_l \in \mathcal{S}_r\}$.

Here, we convert strings that label nodes into vectors of identities in the HIBE scheme. To encrypt M to identity $\texttt{id}$ with trust rating $R = \sum_{i=0}^{l-1} R_i 2^{-(i+1)}$, we simply encrypt M under identity $(\texttt{id}, R_0, R_1, \ldots, R_l)$ using the encryption algorithm of the HIBE scheme. A recipient in possession of the private key set $\mathcal{D}_{\texttt{id},r}$ with $r \leq R$ can derive an appropriate private key for decryption.

It is easy to see that this construction for TIBE from HIBE will be IND-CCA secure providing the HIBE scheme has the same security strength. It can be instantiated using a variety of HIBE schemes from the literature. Of particular interest for reducing ciphertext expansion is the HIBE scheme of [7, Section 3.2] which offers constant size ciphertexts (3 bilinear group elements), IND-CCA security in the standard model (albeit with an exponentially bad security reduction to a decisional variant of a Bilinear Diffie-Hellman Exponent assumption), pairing-free encryption, and efficient decryption (dominated by the cost of 2 pairings). Unfortunately, the private keys in this HIBE and thus in the resulting TIBE are much larger than in schemes obtained using the methods in the previous section.

5. DISCUSSION

In this section we briefly examine the benefits of our combined trust and key encryption approaches, using Mobile Ad Hoc Networks (MANETs) as a concrete example. In addition we highlight some outstanding issues not immediately addressed by our system.

The challenge of designing a lightweight security infrastructure to enable secure information sharing amongst coalition partners presents many complex problems, particularly in the MANET environments that will be seen in future military and emergency response networks. These networks are intended to be self-organising, self-discovering and capable of rapid changes in both organisational and network topology. Additionally, network elements (nodes) within these networks are likely to be severely resource-constrained, particularly in terms of bandwidth and battery capacity.

As communication costs dominate computation costs in ad hoc networks (by several orders of magnitude) [12, 24, 27], our TIBE approach is particularly amenable to the constraints these networks impose. The low communication overhead, coupled with the ability to have an off-line TA (for all purposes except the distribution of private keys) is an attractive property in such networks. Furthermore, to avoid a TA becoming a single point of vulnerability/failure in a network, a TIBE TA can be easily distributed in a threshold manner using standard techniques, see for example [18, 29].

Additionally, as a by-product of our economic model, the threat posed by node replication [39] and Sybil attacks [19] can be significantly reduced in our system. As an entity pays for new keys out of its risk budget, a compromised, or Sybil node, must distribute its wealth amongst its replicants/Sybils, thus significantly reducing the viability of each subsequent false replicant/Sybil.

5.1 Issues

Bootstrapping: The selection of a bootstrapping mechanism for establishing trust with unknown entities will largely be dictated by the environment in which our system must operate. At an organisational level bootstrapping may be carried out through contractual agreements where there is clear liability in the event that something bad happens. In a MANET environment, we may adopt one of a number of pre-existing proposals for bootstrapping trust. For example, bootstrapping trust may occur through one of the following: establishing trust through physical contact [42, 3], establishing trust through negotiation [32, 31] or, alternatively, by establishing trust based on the average benefit obtained from previous first-time interactions [30].

Payments: The issues surrounding the procurement and trading of risk tokens used to purchase keys falls outside of the immediate scope of our work. For this paper, we have assumed the existence of a remuneration scheme in which entities are pre-assigned a risk budget where the value of this budget may, for example, be based on an entities role within an organisation. Remuneration, and the trading of risk tokens may occur via credit-card like processing supported by a payment clearing infrastructure, or in a peer-to peer manner between entities, as per [48, 11, 13, 14], or, alternatively, via providing evidence of successful interactions to a third party, as per [50, 1, 20, 51]. Much like bootstrapping trust, the selection of an appropriate payment mechanism will be dictated on the environment in which our system will operate.

6. CONCLUSIONS

We are beginning to see new computing paradigms emerging around the concept of intra- and inter-organisational information sharing. In a Grid environment, a virtual organisation is established to share data and resources amongst a set of independent organisations. Likewise in MANETs, multiple organisations may come together to form a coalition to satisfy some transient objective. In both instances, getting the right information, to the right people, at the right time, without a loss of governance remains a challenging problem.

Towards this grand challenge, this paper has presented a novel trust and key management paradigm for securing pan-organisational information flows. We have described a realisation of our trust management paradigm that is built around an economic model and a number of trust-based cryptographic primitives. The proposed trust management paradigm does not rely on collective opinions to estimate the trust score for an entity, thereby making it resilient to shilling and collusion attacks.

In this paper, we have presented a detailed analysis of our economic model and have shown that: (i) it is indeed possible to derive pricing models wherein entities are incentivised to report their honest trust scores to a trust authority, (ii) using the last two monitoring time windows is necessary and sufficient for trust management, and (iii) a finite lag in monitoring does not impact long run information leakage rate. We have also presented a suite of trust-based encryption techniques that complement our economic model. We have identified computation and communication tradeoffs between the proposed encryption techniques and described potential applications across scenarios ranging from coalition MANETs to the wired Internet.

7. REFERENCES

[1] L. Anderegg and S. Eidenbenz. Ad Hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. In *Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom 2003)*, pages 245–259. ACM, 2003.

[2] G. Athanasiou, L. Tassiulas, and G. S. Yovanof. Overcoming Misbehaviour in Mobile Ad Hoc Networks: An Overview. *Crossroads The ACM Student Magazine*, (114):23–30, 2005.

[3] D. Balfanz, D.K. Smetters, P. Stewart, and H.C. Wong. Talking to strangers: Authentication in ad-hoc wireless

networks. In *Proceedings of Network and Distributed System Security Symposium 2002 (NDSS'02)*, San Diego, CA, February 2002.

[4] S. Balfe, K. Boklan, Z. Klagsburn, and K.G. Paterson. Key refreshing in identity-based cryptography and its applications in MANETs. In *Proceedings of the 2007 IEEE Military Communications Conference (Milcom 2007)*, 2007.

[5] M. Barbosa and P. Farshim. Efficient Identity-Based Key Encapsulation to Multiple Parties. In *Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, 2005.

[6] K. Bentahar, P. Farshim, J. Malone-Lee, and N.P. Smart. Generic Constructions of Identity-Based and Certificateless KEMs. *Journal of Cryptology*, 21(2):178–199, 2008.

[7] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical Identity-Based Encryption with Constant Size Ciphertext. In R. Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005.

[8] D. Boneh and M.K. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[9] D. Boneh and B. Waters. Conjunctive, Subset, and Range Queries on Encrypted Data. volume 4392 of *LNCS*, pages 535–554. Springer, 2007.

[10] S. Buchegger and J.-Y. Le Boudec. Self-Policing Mobile Ad Hoc Networks by Reputation Systems. *Communications Magazine, IEEE*, 43(7):101–107, 2005.

[11] L. Buttyán and J.-P. Hubaux. Enforcing Service Availability in Mobile Ad-Hoc WANs. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2000)*, pages 87–96. IEEE Press, 2000.

[12] L. Buttyán and J.-P. Hubaux. Stimulating cooperation in self-organising mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5):579–592, October 2003.

[13] L. Buttyán and J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Network Applications*, 8(5):579–592, 2003.

[14] L. Buttyán and M. Jakobsson. Node Cooperation in Hybrid Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 5(4):365–376, 2006.

[15] H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy (S&P 2003)*, pages 197–213. IEEE Computer Society, May 2003.

[16] P.-C. Cheng, P. Rohatgi, C. Keser, P.A. Karger, G.M. Wagner, and A.S. Reninger. Fuzzy Multi-Level Security: An Experiment on Quantified Risk-Adaptive Access Control. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP 2007)*, pages 222–230. IEEE Computer Society, 2007.

[17] M. Conti, E. Gregori, and G. Maselli. Cooperation Issues in Mobile Ad Hoc Networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW 2004)*, pages 803–808. IEEE Computer Society, 2004.

[18] H. Deng, A. Mukherjee, and D. P. Agrawal. Threshold and Identity-based Key Management and Authentication for Wireless Ad Hoc Networks. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2004)*, pages 107–111. IEEE Computer Society, 2004.

[19] J.R. Douceur. The Sybil Attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems IPTPS 2001)*, pages 251–260. Springer-Verlag, 2002.

[20] S. Eidenbenz, G. Resta, and P. Santi. COMMIT: A Sender-Centric Truthful and Energy-Efficient Routing Protocol for Ad Hoc Networks with Selfish Nodes. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2005)*, 2005.

[21] L. Eschenauer and V.D. Gligor. A Key-Management Scheme for Distributed Sensor Networks. In *Proceedings of the 9th ACM conference on Computer and communications security (CCS 2002)*, pages 41–47. ACM, 2002.

[22] P. Farshim. *Extensions of Public-Key, Identity-Based and Certificateless Encryption Schemes*. PhD thesis, University of Bristol, 2008.

[23] C. Gentry and A. Silverberg. Hierarchical ID-Based Cryptography. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.

[24] A. Ghose, J. Grossklags, and J. Chuang. Resilient data-centric storage in wireless ad-hoc sensor networks. In M.-S. Chen et al., editor, *Proceedings of the 4th International Conference on Mobile Data Management (MDM 2003)*, pages 45–62. Springer-Verlag, Jan 2003.

[25] Q. He, D. Wu, and P. Khosla. SORI: A Secure and Objective Reputation-Based Incentive Scheme for Ad-Hoc Networks. In *Proceedings of the 3rd IEEE Wireless Communications and Networking Conference, (WCNC 2004)*, pages 825–830. IEEE Press, 2004.

[26] K. Hoeper and G. Gong. Bootstrapping Security in Mobile Ad Hoc Networks Using Identity-Based Schemes with Key Revocation. Technical Report CACR 2006-04, Centre for Applied Cryptographic Research (CACR) at the University of Waterloo, Canada, 2006.

[27] D. Hwang, B.-C. C. Lai, and I. Verbauwhede. Energy-memory-security tradeoffs in distributed sensor networks. In *Ad-Hoc, Mobile, and Wireless Networks: Third International Conference, ADHOC-NOW*, pages 70–81. Springer-Verlag, 2004.

[28] S. Kamvar, M. Schlosser, and H. Garcia-Molina. EigenTrust: Reputation management in P2P networks. In *Proceedings of the 12th World Wide Web Conference (WWW 2003)*, 2003.

[29] A. Khalili, J. Katz, and W.A. Arbaugh. Toward Secure Key Distribution in Truly Ad-Hoc Networks. In *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT 2003)*, pages 342–347. IEEE Computer Society, 2003.

[30] K. Lai, M. Feldman, I. Stoica, and J. Chuang. Incentives for cooperation in peer-to-peer networks. 2003.

[31] J. Li, N. Li, and W.H. Winsborough. Automated trust negotiation using cryptographic credentials. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 46–57. ACM, 2005.

[32] N. Li and W. Winsborough. Towards Practical Automated Trust Negotiation. In *Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 2002)*, page 92. IEEE Computer Society, 2002.

[33] C.D. McCollum and J.R. Messing L. Notargiacomo. Beyond the Pale of MAC and DAC-Defining New Forms of Access Control. In *Proceedings of the 1990 IEEE Symposium on Security and Privacy (S&P 1990)*, pages 190–200. IEEE Computer Society, 1990.

[34] P. Michiardi and R. Molva. CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In B. Jerman-Blazic and T. Klobucar, editors, *IFIP TC6/TC11 6th Joint Working Conference on Communications and Multimedia Security*, volume 228 of *IFIP Conference Proceedings*, pages 107–121. Kluwer Academic, 2002.

[35] A.C. Myers and B. Liskov. A Decentralized Model for Information Flow Control. In *Proceedings of the 1997 Symposium on Operating Systems Principles (SOSP 1997)*, pages 129–142. ACM Press, 1997.

[36] A.C. Myers and B. Liskov. Complete Safe Inforamtion Flow with Decentralized Labels. In *Proceedings of the 1998 IEEE Symposium on Security and Privacy (S&P 1998)*, pages 186–197. IEEE Computer Society, 2001.

[37] Jason Prograrm Office. HORIZONTAL INTEGRATION: Broader Access Models for Realizing Information Dominance. Special Report JSR-04-13, MITRE Corporation, 2004.

[38] D. Page, N.P. Smart, and F. Vercauteren. A Comparison of MNT Curves and Supersingular Curves. *Appl. Algebra Eng., Commun. Comput.*, 17(5):379ï£¡392, 2006.

[39] B. Parno, A. Perrig, and V. Gligor. Distributed Detection of Node Replication Attacks in Sensor Networks. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P 2005)*, pages 49–63. IEEE Computer Society, 2005.

[40] D. Roberts, G. Lock, and D.C. Verma. Holistan: A Futuristic Scenario for International Coalition Operations. In *In Proceedings of Fourth International Conference on Knowledge Systems for Coalition Operations (KSCO 2007*, 2007.

[41] E. Shi, J. Bethencourt, T.-H. Chan, D. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *IEEE Symposium on Security and Privacy*, pages 350–364, 2007.

[42] F. Stajano. The resurrecting duckling - what next? In *Revised Papers from the 8th International Workshop on Security Protocols*, pages 204–214. Springer-Verlag, 2001.

[43] G. Stoneburner, A. Goguen, and A. Feringa. Risk Management Guide for Information Technology Systems. Special Report 800-300, NIST, 2002.

[44] N. Swamy, M. Hicks, and S. Tsang. Verified Enforcement of Security Policies for Cross-Domain Information Flows. In *Proceedings of the 2007 Military Communications Conference (MILCOM 2007)*, pages 192–206. IEEE Computer Society, 2007.

[45] R. Anderson T. Moore, J. Clulow and S. Nagaraja. New Strategies for Revocation in Ad-Hoc Networks. In *Proceedings of the 4th European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS 2007)*, pages 232–246. Springer, 2007.

[46] J.A. Vaughan and S. Zdancewic. A Cryptographic Decentralized Label Model. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P 2007)*, pages 192–206. IEEE Computer Society, 2007.

[47] L. Xiong and L. Liu. Supporting reputation based trust in peer-to-peer communities. In *IEEE Transactions on Knowledge and Data Engineering (TKDE), Special Issue on Peer-to-Peer Based Data Management, 16(7)*, July 2004.

[48] B. Yang and Hector Garcia-Molina. Ppay: micropayments for peer-to-peer systems. In *Proceedings of the 10th ACM conference on Computer and communications security (CCS '03)*, pages 300–310. ACM, 2003.

[49] S. Zdancewic and A.C. Myers. Secure Information flkows and CPS. In *Proceedings of the 10th European Symposium on Programming (ESOP 2001)*, pages 46–61. Springer, 2001.

[50] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, pages 1987–1997. IEEE Press, 2003.

[51] S. Zhong, L.E. Li, Y.G. Liu, and Y.R. Yang. On Designing Incentive-Compatible Routing and Forwarding Protocols in Wireless Ad-Hoc Networks: An Integrated Approach Using Game Theoretic and Cryptographic Techniques. *Wireless Networks*, 13(6):799–816, 2007.