Utility Sampling for Trust Metrics in PKI

Dakshi Agrawal* and Charanjit Jutla*

IBM T. J. Watson Research Institute,
Yorktown Heights, NY 10598-704

Abstract. We propose a new trust metric for a network of public key certificates, e.g. as in PKI, which allows a user to buy insurance at a fair price on the possibility of failure of the certifications provided while transacting with an arbitrary party in the network. Our metric builds on a metric and model of insurance provided by Reiter and Stubblebine [8], while addressing various limitations and drawbacks of the latter. It conserves all the beneficial properties of the latter over other schemes, including protecting the user from unintentional or malicious dependencies in the network of certifications. Our metric is built on top of a simple and intuitive model of trust and risk based on "utility sampling", which maybe of interest for non-monetary applications as well.

Keywords: Public Keys, PKI, Certificates, Trust, Authentication, Insurance, Beta Densities.

1 Introduction

In the rapidly growing world of Internet and e-commerce, trust in entities, especially business entities with which a user may have monetary transactions, is a challenging issue. Consider the case of PKI (public key infrastructure), where a reputed entity called *certificate authority* (CA) authenticates (usually via a chain of public key authentications [3]) a target entity's public key and other information about the target entity. This however, requires an implicit (user's) trust in the CA's ability to associate the "other information" with the target public key. As the authentication of the target public key is done via a chain of authentications, a similar trust in the ability of intermediaries in associating information about the next entity in the chain with the public key of the next entity

* Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

is required. On top of this, the user must trust CA's public key. Although, this last trust may be justified, hinted by the fact that the CA is a reputed entity, the trust in other entities in the chain of authentications is never perfect, and this is especially so in the last link.

Thus, a user is inevitably led to determine the net trust it can place in the full chain of authentications, usually by various algebras on trust metrics. Moreover, this net trust, or the utility of this trust, also depends on what information was authenticated. For example, if the information in a certificate authenticates a public key with a business name and its web address, then the user's net trust determined by such algebras is only about the public key being associated with the business name. The user must independently determine how much it trusts the business name being a bona-fide business. On the other hand, if the certificate also authenticates, along with the above information, that the business has a five star service quality, then the user is more inclined to trust the business.

Although various trust metrics and algebras for networks of links have been studied [10, 6, 2, 9, 5], our focus is on trust metrics for network of authentications as in PKI. Even in the context of PKI, various trust metrics have been proposed. Of particular importance is the work of Reiter and Stubblebine [8], who proposed a set of eight principles which a good trust metric must follow. They also proposed a metric which claimed to follow these principles. As one of the more important principles they had required that *the metric's output should be intuitive and relevant to the authentication decision.* In their solution a metric can be computed which represents the amount for which the information bound to the target public key is *insured* for.

Although, this metric is of great relevance to the user, and also remedies many other problems with earlier metrics, it has a few major drawbacks: (a) it does not tell the user what the premium for the insurance is, (b) it does not model how the insurance amounts were determined, and (c) it forces the user to deal with all entities in the network for insurance claims. Although, the drawbacks (b) and (c) may really be implementation issues, we believe the drawback (a) is of real concern, and actually violates one of their principles, namely "the metric should take into account as much information as possible that is relevant to the authentication decision that the user is trying to make".

In our solution, which builds on Reiter and Stubblebine's insurance metric, not only does the final metric allow the user to determine the insurance premium, it also allows the user to insure any an amount to his or her choice, upto a limit. Further, this limit is part of the metric. Moreover, the user needs to interact with only one entity (usually the top level CA) for insurance claims and premiums. We also model the trust metrics on a simple and intuitive theory of "*utility sampling*", which maybe of independent interest.

1.1 Examples Involving the New Metric

Our scheme is best illustrated with a sequence of examples, each more complicated than the previous. A sequence of authentications starting with a CA (say, with public key K_S), and ending with a target public key K_T, is represented by a line graph, with nodes being the public keys of the entities involved, and the directed edges representing signatures. Each edge, say from K_A to K_B, is also labelled with information which is associated with K_B, and signed with secret key corresponding to K_A. For example, this information may include the name B of the business associated with key K_B. The information also includes a pair of values $p(K_A, K_B)$ and $c(K_A, K_B)$, called the **probability of accuracy**, and the **confidence** respectively. We will drop K_A and K_B, when they are clear from context. The value p represents the probability, according [2] to A, that the information about K_B is reliable, including K_B's owner's ability to recommend other entities. In absence of any other information then, conditioned on the fact that owner A of K_A is fully reliable, $p(K_A, K_B)$ represents the best estimate about the probability of accuracy of information associated with K_B. If the line graph has an edge from K_S to K_A, then $p(K_S, K_A)p(K_A, K_B)$ is the probability (according to S) of B being reliable (in absence of any other paths of edges between S and B). In this manner, S can calculate, the probability (according to S itself) that the target T associated with K_T, and its information is reliable. Using this probability, it can calculate the premium it will charge to insure a certain amount of transaction with T. More details about this calculation and the underlying model will be given later.

However, S cannot insure an unlimited amount using this probability information, and here is where the confidence c comes in. As we will see later, the probability $p(K_A, K_B)$ is estimated by A, using what we call *utility sampling*. In simple terms, $c(K_A, K_B)$ is the total dollar amount of transactions which have been involved using that edge. Hence, we require that this edge can only be used in an insurance of upto that amount (or possibly, a certain function of it). In a line graph, the minimum c value on the edges of the line graph determines the maximum amount that can be insured. For example, in the line graph in fig 1, represented by nodes K_S, K_A, K_B and K_T, the maximum amount insurable is 100.

The situation gets more interesting when we have multiple paths from S to the target entity T. As an example, again consider fig 1. Any path P_i from S to T can be treated as a line graph, and the min value of c on P_i, denoted c_i, is the limit on the amount insurable using that path. For instance, using the middle path K_S, K_A, K_B, K_T, the maximum insurable value is 100, and the corresponding probability is $(1*0.98*0.99)$. On the other hand, using the top path K_S, K_C, K_D, K_T, the maximum insurable value is 200, with a corresponding probability of $(0.99*0.99*0.90)$. Now, if a user is interested in a transaction of amount only 100 with T, then he could be offered insurance

[2] When we say that the probability is according to A, it means that A has determined an underlying probability space, and it is in this space that A has estimated the probability. We will address these issues later in more detail.

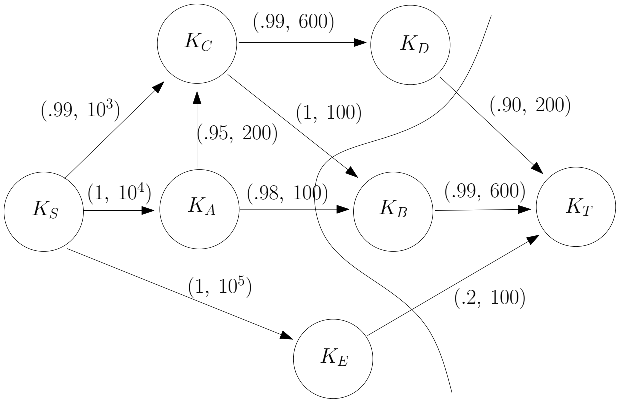

Fig. 1. The New Metric

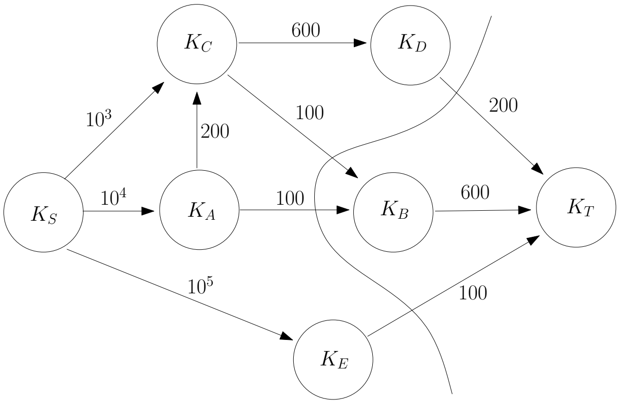

Fig. 2. The Reiter-Stubblebine Metric

based on any of these paths used to calculate the premium (or the average premium – such more complicated strategies are considered and modeled later in the paper). On the other hand, if he is interested in a transaction worth 300, the first 200 can be filled using the top path and the premium determined using the corresponding probability, and the remaining 100 filled using the middle path (or some other path). In general, the total insurable amount equals the *max flow* in the graph (using the c labels), and this corresponds to the metric developed in [8] by the famous max-flow min-cut theorem [4, 7].

One potential problem with various other metrics (e.g. as in PGP [10]) is that some of the paths and trust values may be dependent. For example, in fig 1, owner of K_C (owner of K_D) may be same as K_A (same as K_B resp.), maliciously or otherwise. In our model, the end user does not care, as it gets its insurance at a fair premium (the premium is fair as there may be several CA's providing a market). It is upto S to determine that the owners of K_A and K_C (say, A and C) are same or different. Even if it does determine that they are same, as we will see later, our model incorporates the fact that S has assessed both A and C separately while obtaining the values $(p(K_S, K_A), c(K_S, K_A))$ and $(p(K_S, K_C),\ c(K_S, K_C))$.

The rest of the paper is orgnaized as follows. We first describe the Reiter-Stubblebine Insurance model and metric, along with its advantages over other metrics. We also discuss its drawbacks. In section 3 we describe our model and the new metric. The section also describes the algorithms for calculating the metrics. In section 4 we describe the underlying probability model, and the notion of utility sampling which is used to justify our metric.

2 The Reiter-Stubblebine Model and Metric

We first describe the Reiter Stubblebine (RS-) metric [8]. The metric operates on a directed (acyclic) graph. The nodes in the graph are public keys, and the edge $K_A \rightarrow K_B$ exists in the graph if the *user* is in possession of a certificate that assigns attributes (including an owner) to K_B, and whose signature can be verified using K_A. Each edge is labeled with the attributes included in the certificate that the edge represents. It is assumed that the attributes in a certificate are consistent.

Each edge $K_A \rightarrow K_B$ also has a numeric label that represents the amount of money for which the owner of K_A insures the attributes and integrity of K_B. In other words, it is the value for which the owner of K_A will be liable to the user if the attributes bound to K_B in the certificate are incorrect, or if the private key (corresponding to K_B) is used to mislead the user, intentionally or otherwise. It is also natural to assume that the numeric label is part of the certificate.

The metric is best described using the example in fig 2. If the attributes bound to K_T (the target public key) turn out to be false, the owners of K_D, K_B, and K_E are each liable to the user for the amount of 200, 600 and 100 respectively. It is also possible that when the user goes to, say owner of K_B, the owner (or its attributes as certified) turns out to be delinquent, and hence the user now is owed by owners of K_A and K_C, for the amount of 100 each.

Reiter and Stubblebine show that, in case of a false binding for the target key, the *minimum insured amount* is equal to the *minimum capacity cut* in the graph from K_S to K_T. For example, in fig 2, the minimum cut as shown has value 500, and that is the minimum insured value.

Some of the salient qualities of this metric and model (following the principles enunciated in [8]) are:

1. the user is not required to ascertain name to key bindings to construct the model, except for the root CA, whose name to key binding is reputed,
2. the final metric computed is intuitive,

3. the final metric lets the user ascertain the risk involved in using K_T,
4. the final metric is computed easily using the Ford-Fulkerson algorithm [4] (or see [7]),
5. the metric can be computed with partial information, and still give meaningful results,
6. the insurance metric allows the user to be protected from dependencies in the graph, whether they are intentional or malicious. This is an important differentiator from other metrics, e.g. the metric used in PGP [10]. In PGP, a target key binding is deemed reliable if two marginally reliable keys are authenticating it. However, those two keys may belong to the same marginally reliable person (i.e. the edges are highly dependent). In Reiter Stubblebine, if there were two such authentications from the same person, the person is liable for both edges in case of a failure.

Before we look at the drawbacks of this model and metric, we point out that [8] does mention that the metric could also include trust values, and then use only those edges which transcend a certain trust value. However, no specifics are given. The authors also mention that their metric does not address some real world issue, e.g. the insurance premium, determining liable parties, and recovery of funds. Indeed, these are some of the major drawbacks of this metric. In fact, since the metric just computes a total insured value, regardless of the value of the transaction, the user is left to ascertain the risk involved in undertaking this transaction. One could argue that the user should not undertake the transaction if it is of value a certain multiple of the insured value, and should go ahead and do the transaction if the value is less than this threshold. Clearly in such a situation, the insurance premium for the different valued transactions has to be different. For example, if the minimum insured value turns out to be a $10,000, then if the user is only doing a transaction worth $1000, he should need to pay a much lower premium. This calculation can possibly be done using a fixed rate, but then the question arises as to who the premium goes to, and how is it split among different entities? The factors involved in calculating the premium, must definitely involve risk factors and trust values of individual keys (and their owners), and that should be part of the model.

To drive home this point, consider the link from K_S to K_A, which has a value of 10^4. Presumably, owner S of K_S has determined that its attestation of binding of K_A to its owner is highly reliable, and hence S can offer insurance of 10^4. For instance, K_A itself may be a reputed computer supplier, with highly secure web servers. Further, K_A itself is attesting other smaller vendors which provide accessories etc, for example K_B and continuing further K_T. Now, even though K_A's owner is reliable (as a service provider and in protecting its keys), there is a small, even though minuscule, chance that it may fail. Let's say the probability of that happening is 1 in a million transactions. But, when it happens, S is liable for 10,000 to each user, even if they were only involved in 100 dollar transactions with T. There maybe 100,000 such users who had signed up for this insurance (hopefully paying no more than a dollar for insurance premium to all entities combined).

The solution we provide actually resolves these problems, while providing a sound and intuitive underlying model of trust.

3 The New Model and Metric

Our model shares many properties with the Reiter-Stubblebine model, and can in fact be seen as an extension, although there are fundamental differences. As in the RS-model, the metric operates on a directed acyclic graph, with the nodes in the graph being public keys. An edge $K_A \rightarrow K_B$ exists in the graph if the user is in possession of a certificate that assigns attributes (including exclusive owner(s)) to K_B, and whose signature can be verified using K_A. Each edge is labeled by the attributes included in the certificate that the edge represents. Each edge is also labeled with two numeric values p and c (whose significance will be pointed out later).

There maybe multiple source nodes in the graph, and we will only be interested in those source nodes in which the user has complete trust, including the binding of information to the source public key. Modeling limited trust in source nodes is beyond the scope of this paper. The various source nodes can be seen as providing alternative metrics of use to the end user, and possibly a market for the metrics.

In the graph, there is a target node, say K_T, and the end user is interested in a transaction with the owner of K_T. The ultimate goal of the metric calculation is for the end user to determine the risk involved in the transaction, or alternatively in the trust to place in the transaction.

We assume that all the attributes in any certificate are consistent. For simplicity, we assume that for each node, the information attributed to it (other than p and c) by various incoming edges is the same. This is not a limitation of our model, but is there only to simplify the exposition. Thus we can assume that the attributes are labeled on the nodes, whereas the p and c values are labels on the edges. Moreover, we assume that at the end of each transaction it can be determined (unambiguously for the end user and the source node) whether any of the attributes in the certificates involved were falsified. Here are a few examples of attributes in certificates:

- Web Address = www.xxx.com,
- Name and Physical Address = N and A,
- Dun and Bradstreet Solvency Rating = A (i.e. third party rating value),
- The business associated with this public key WILL deliver.

For each transaction with owner of K_T, and each source node K_S, the metric provides a limit l on the amount of insurance that owner of K_S is willing to provide to the end user, and for each amount m less than l, the metric provides a premium for which the owner of K_S is willing to sell insurance to the end-user for an amount of m.

Note that, even if the end user is not interested in buying the insurance, or even if K_S is not in the business of selling insurance, the metric can be calculated by all parties regardless, and hence provides a good metric of the risk involved in the transaction. Of course, as we will see later, it can be argued that the values p and c reflect actual values more accurately, if monetary transactions or reputations were involved.

3.1 The Metric Calculation Algorithm

We start by describing the *algorithm to compute the upper limit* on the amount insurable. This value is calculated based solely on the c labels of the edges. This upper limit is the maximum K_S-K_T flow in the directed graph [4, 7], with the capacity on each edge being the c value of the edge. By the famous max-flow min cut theorem [4], this maximum flow is equal to the minimum capacity K_S to K_T cut of the c-labeled graph. For example in fig 1, the min cut is as depicted by the curved line, and its value is 500, which is also the max flow in the graph. These values can be computed efficiently using the Ford-Fulkerson max-flow algorithm [4, 7]. There can be variations on this limit calculating algorithm, where the maximum amount insurable is some function of the max flow, as the model on which this algorithm is based is intuitive, and there is no hard reality on which it can be based.

As for the intuition behind this algorithm, we first describe what the c values on the edges are supposed to represent. The c value on each edge $K_A \rightarrow K_B$ is the total dollar amount of transactions that have ever been insured involving this edge. We will discuss the "involvement" aspect later in more detail. We will also relate $1/c$ to the variance in the probability estimates made by owner of K_A about the attributes of K_B in section 4. In effect, if the c value is too low, the variance in the probability estimate is so high that it is not practical to determine a good premium for insurance. So, a natural upper bound on the insurance amount is c, or some multiple of it. Hence, for any path from K_S to K_T, the maximum amount insurable should be the minimum c value on that path. By the same reasoning, the maximum amount insurable using the whole directed graph should be the maximum flow in the graph (with capacity bound on each edge being its c label).

Note that if the amount to be insured is much smaller than the maximum amount insurable, one may choose a variation of the algorithm, where not all paths are considered, and a rough estimate on the premium suffices, usually with the benefit of the doubt going to the end user. This will become more apparent when we describe the premium calculation algorithm, which comes next. In general, the premium calculation algorithm can become a complex optimization problem. But, for most purposes simplicity in the algorithm maybe more of a determining choice than the optimal premium value (whether it is optimal from the end user's perspective or K_S's perspective).

So a simple variation of the **premium calculating algorithm** does the following. Assume that the maximum flow f has been determined (based on the c labels), and all the paths with non-zero flow are determined. For each such path P_i, lets say the flow attributed to the path is f_i, with the sum being f. For each path P_i, calculate the probability of accuracy p_i to be the product of the p labels on edges on the path. For example, in fig 1, if the path is K_S, K_A, K_B, K_T, then the probability of accuracy of this path is $1 * 0.98 * 0.99$. Let's assume that for each dollar amount to be insured using this path P_i, there is a formula for calculating the premium based on the probability p_i. For example, let s_i be the price (selling price) of the premium, and a_i be the amount owner of K_S (say S) pays to the owner of K_A (say A) for insuring δ_i fraction of the amount. Also, assume that S expects a γ fraction expected profit per transaction. Then,

$$(s_i - a_i)(1 - p_i) - (1 - \delta_i)p_i = \gamma$$

The value a_i is determined inductively, and in fact towards the bottom end of the chain of authentication edges, the δ_i value could be zero. For instance in our example, A may not buy further insurance from B, and B may just be in a service contract with A. In case of failure of B or some node below B, the reputation of B goes down and he/she risks losing the service contract. There are various other ways the calculation of the premium s_i can be done based on p_i and the path P_i, and these methods are not the main focus of this paper.

Thus, for each path P_i, we have the maximum amount insurable, i.e. f_i, and the rate of insurance premium s_i. The rest of the algorithm is straightforward. First, the paths are sorted in increasing order of rate s_i. If the user wants to insure an amount u ($< f$), then the first f_1 dollars out of u are insured according to P_1 using rate s_1, and the next f_2 dollars are insured using path P_2 using rate s_2, and so forth till all the u dollars are filled.

Alternatively, the first f_1 dollars could be insured at the average premium rate (over all i), and the next f_2 dollars can be insured at the average premium rate (excluding s_1), and so forth. This alternative represents the estimates of risk more accurately, especially if there are negative recommendations (e.g. p values less than 1/2).

3.2 Updating the p and c Values

The way the paths P_i are used to calculate the requisite premiums, provides a way for the entities to improve their estimates on the probability of accuracy p, as well as c. This will become more apparent in section 4 where we model these probabilities. Thus, if a path P_i was indeed used to provide insurance for amount f_i, each edge in the path can use this as a sample of size f_i for its probability estimation. Note that there is no hard and fast rule, and the entities may have their own way of weighting positive and negative samples. In fact, the entities may have their own completely different estimation mechanism, and we provide the mechanism only as a guide.

While the node attesting the target node, maybe estimating the probability of binary hypothesis, the intermediate nodes are attesting the probability of success itself of the next node. For example, in fig 1, the node K_A is estimating the accuracy of B to be 0.98. This accuracy of B includes its attestation that owner of K_T is reliable with probability 0.99. However, the estimation of $p(K_A, k_B)$ can be done as follows. Let $K_{T1}, ..., K_{Tn}$ be the list of all public keys (and their attributes) that B is attesting, and let the (p, c) values estimated by B about K_{Ti} be (p_i, c_i). Then after a time period, where for each i, A attested T_i via B using $\mathit{\Delta} c_i$ additional samples (dollar amounts), and $\mathit{\Delta} r_i$ of the samples showed positive behavior by T_i, the new value of $p(K_A, K_B)$ (in short p_A^{new}) is given by

$$p_A^{\mathrm{new}} = \frac{p_A^{\mathrm{old}} + \sum_i \mathit{\Delta} c_i \cdot (1 - |1 - \frac{\mathit{\Delta} r_i p_i}{\mathit{\Delta} c_i}|)}{c_A^{\mathrm{old}} + \sum_i \mathit{\Delta} c_i}$$

4 Utility Sampling

We start by describing second order probabilities. Let h be a binary hypothesis that is true with probability $p \in [0, 1]$. A second order probability (SOP) density function f is a function such that

$$f(p) \geq 0$$

$$\int_0^1 f(p)dp = 1$$

Thus a second order probability density function is a density on the first order probability p. SOP density functions are used in modeling situations where the first order probability p is not known precisely.

Assume that to start with, an agent A does not have any experience with the hypothesis h. We can model this by assuming that for the agent A the *a priori* distribution, $\psi_0(p)$ on p is uniform. If the agent A then makes r_1 positive and s_1 negative observations of the hypothesis h, then the posteriori probability density of p is given by (see Appendix A)

$$\psi_1(p = \theta | r_1, s_1) = \frac{\mathit{\Gamma}(r_1 + s_1 + 2)}{\mathit{\Gamma}(r_1 + 1)\mathit{\Gamma}(s_1 + 1)}\, \theta^{r_1}(1 - \theta)^{s_1} \tag{1}$$

where $r_1, s_1 \geq 0$ and $\mathit{\Gamma}$ is the gamma function. It is not difficult to derive (see Appendix A) that if the agent subsequently makes r_i positive and s_i negative observations of the hypothesis h in the time epoch i, then the posteriori probability density of p is given by

$$\begin{aligned} &\psi_i(p = \theta | r_j, s_j, 1 \leq j \leq i) \\ &= \frac{\mathit{\Gamma}(r + s + 2)}{\mathit{\Gamma}(r + 1)\mathit{\Gamma}(s + 1)} \theta^r (1 - \theta)^s \end{aligned} \tag{2}$$

where $r = \sum_{j=1}^{i} r_j$ and $s = \sum_{j=1}^{i} s_j$. We make several observations here. First, the posteriori probability densities $\psi_i(\dot{})$ are beta densities [1]. Second, the total number of observations $r + s$, and the fraction $\frac{r}{r+s}$ form a sufficient statistics for estimating the parameter p. And finally, the mean of p is given by

$$\begin{aligned} & E[p|r_j, s_j, 1 \leq j \leq i] \\ & = \int_0^1 \theta \psi_i(\theta|r_j, s_j, 1 \leq j \leq i) d\theta \qquad (3) \\ & = \frac{r+1}{r+s+2} \qquad (4) \end{aligned}$$

while the variance of p is given by

$$\begin{aligned} & V[p|r_j, s_j, 1 \leq j \leq i] \\ & = \frac{(r+1)(s+1)}{(r+s+2)^2(r+s+3)} \end{aligned}$$

In our model of network of authentications, we annotate each edge of the graph by the pair $(p, c) = (\frac{r+1}{r+s+2}, r+s)$. It is easy to verify that (p, c) together form sufficient statistics to specify the posteriori density of p. This notation has an intuitive appeal—the first element of the pair denotes the mean of the SOP density function, and the second element denotes the total number of experiences, positive or negative. Note that as $c = (r+s)$ increases the variance goes down, and hence better is our confidence in the first element of the pair. We also note that for large values of r and s, the first parameter can be approximated by $r/(r+s)$.

These two parameters can be readily estimated in different circumstances. For example, if two agents are engaged in monetary transactions, then the second parameter can be put equal to the total amount of money involved in these transactions, while the first parameter can be put equal to the fraction of money involved in transactions whose outcome was *positive* (as defined by the concerned agent), and hence the name *utility sampling*.

References

1. W. H. Beyer, *CRC Standard Mathematical Tables*, 28th ed. Boca Raton, FL: CRC Press, pp. 534-535, 1987.
2. T. Beth, M. Bercherding, and B. Klein, “Valuation of trust in open networks”, Computer Security - ESORICS 94, LNCS 875.
3. A. D. Birrell, B. W. Lampson, R. M. Needham and M. D. Schroeder, “ A global authentication service without global trust”, Proc. IEEE Symp. on Security and Privacy, April 1986.
4. L.R. Ford, Jr. and D. R. Fulkerson, “Maximum flow through a network”, *Canadian Journal of Mathematics*, 8:399-404, 1956.
5. V. Gligor, S. Luan and J. Pato, “On inter-realm authentication in large distributed systems”, Proc. IEEE Symp. on Security and Privacy, May 1992.

6. U. Maurer, “Modeling a public key infrastructure”, Coputer Security - ESORICS 96, LNCS 1146.

7. C. Papadimitriou and K. Steiglitz, “Combinatorial Optimization: Algorithms and Complexity”, Prentice Hall, 1982.

8. M. Reiter, S. Stubblebine, “Toward Acceptable Metrics of Authentication”, Proc. IEEE Symp. on Security and Privacy, 1997.

9. M. Reiter, S. Stubblebine, “Path independence for authentication in large-scale systems”, Proc. ACM Conference on Computer and Communications Security, April 1997.

10. P. Zimmerman, *PGP User's Guide*, Colume I and II, Oct. 1994. Included in the PGP 2.6.2 distribution.

A Beta Distributions and Their Properties

The following integral plays a crucial part in deriving many properties of the beta distribution.

$$\begin{aligned}
&\int_0^1 x^r(1-x)^s dx \\
&= \frac{s}{r+1}\int_0^1 x^{(r+1)}(1-x)^{(s-1)}dx \\
&= \frac{s!}{(r+1)(r+2)\ldots(r+s)}\int_0^1 x^{(r+s)}dx \\
&= \frac{s! \quad r!}{(s+r+1)!} \\
&= \frac{\Gamma(s+1)\Gamma(r+1)}{\Gamma(s+r+2)} \qquad (5)
\end{aligned}$$

In the above derivation, second step follows from integration by parts by making a substitution $u = (1-p)^s$ and $v = \frac{p^{r+1}}{r+1}$.

Using (5), we can derive mean and variance of beta densities. Specifically, the mean of the SOP density function $\psi(p|r,s)$ is given by

$$\begin{aligned}
E[p] &= \int_0^1 \theta \ \psi(\theta|r,s)d\theta \\
&= \int_0^1 \theta\frac{\Gamma(r+s+2)}{\Gamma(r+1)\Gamma(s+1)}\theta^r(1-\theta)^s d\theta \\
&= \frac{\Gamma(r+s+2)}{\Gamma(r+1)\Gamma(s+1)}\int_0^1 \theta^{r+1}(1-\theta)^s d\theta \\
&= \frac{\Gamma(r+s+2)}{\Gamma(r+1)\Gamma(s+1)}\frac{\Gamma(s+1)\Gamma(r+2)}{\Gamma(s+r+3)} \\
&= \frac{r+1}{r+s+2}
\end{aligned}$$

Similarly, we can derive that the variance of a beta distribution is given by

$$\begin{aligned}&V[p|r_j, s_j, 1 \leq j \leq i]\\&= \int_0^1 (\theta - E[p|r_j, s_j, 1 \leq j \leq i])^2 \psi_i(\theta|r_j, s_j, 1 \leq j \leq i) d\theta\\&= \frac{(r+1)(s+1)}{(r+s+2)^2(r+s+3)}\end{aligned}$$

By Bayesian rule of probabilities, we have

$$\begin{aligned}&\psi_i(p=\theta|\mathcal{A}_j, 1 \leq j \leq i)\\&= \frac{\psi_{i-1}(\theta|\mathcal{A}_j \leq j \leq (i-1)) \Pr(\mathcal{A}_i|p=\theta)}{\Pr(\mathcal{A}_i)}\\&= \frac{\binom{r_i+s_i}{r_i}\theta^{r_i}(1-\theta)^{s_i}}{\int_0^1 \binom{r_i+s_i}{r_i} x^{r_i}(1-x)^{s_i} dx}\\&= \frac{\binom{r_i+s_i}{r_i}}{r_i+s_i+1}\theta^{r_i}(1-\theta)^{s_i}\\&= \frac{\Gamma(r_i+s_i+2)}{\Gamma(r_i+1)\Gamma(s_i+1)}\theta^{r_i}(1-\theta)^{s_i}\end{aligned}$$

This article was processed using the LaTeX macro package with LLNCS style