Using Fraud Trees to Analyze Internet Credit Card Fraud

Clive Blackwell

Chapter 2

USING FRAUD TREES TO ANALYZE INTERNET CREDIT CARD FRAUD

Clive Blackwell

Abstract Because of the difficulties inherent in accurately identifying individuals on the Internet, online merchants reduce the risk of credit card fraud by increasing restrictions on consumers. The restrictions are often overly burdensome on consumers and may result in lost sales. This paper uses the concept of a fraud tree, an extension of an attack tree, to comprehensively model online fraud techniques and to suggest defensive obstacles for merchants to counter threats. The fraud tree model can advise merchants about the checks to be performed to reduce risk even in the presence of incomplete knowledge of the circumstances of the transactions. Since fraud cannot be completely avoided, the paper also describes auditing that can be performed to assist merchants in identifying the responsible parties and potentially limiting, if not avoiding, liability due to fraud.

Keywords: Credit card fraud, fraud tree, obstacles, card-not-present transactions

1. Introduction

As more people make purchases online, criminals take advantage of weak authentication checks to commit credit card fraud. The amount of remote fraud, technically called "card-not-present fraud," is estimated to be about £250 million in the United Kingdom – more than all the other types of payment card fraud put together [5].

Merchants are in a difficult position to reduce their liability in a system set up by credit card issuers that emphasizes legal protection for consumers. The fraud tree model presented in this paper is designed to assist merchants. The model uses the concept of a fraud tree, an extension of an attack tree, to comprehensively model online fraud techniques and suggest defensive obstacles for merchants to counter threats. The model can advise merchants about additional checks that can be per-

formed to limit their risk in various transaction scenarios while taking into account the fact that merchants have partial and imperfect knowledge of transactions.

The fraud tree model adapts the anti-goal model provided by the KAOS requirements engineering framework. Following the KAOS requirements, possible fraudulent transactions are determined and appropriate obstacles are proposed. The approach can potentially be applied to other distributed systems where attackers exploit the partial knowledge possessed by system participants, but for which sufficient information can be collected for subsequent attribution.

2. Related Work

Schneier's attack trees [13] provide the foundation for implementing several computer security attack assessment tools. One of the scenarios investigated by Schneier involved attacks against a payment system. However, his work focused on protocol weaknesses instead of the wider perspective taken in this paper. Attack trees have also been used to identify forensic goals [2] and to support investigations of document forgery [3, 4].

The original KAOS framework [16] incorporated a goal model to help determine system requirements and obstacles for analyzing hazards to the goals. An anti-goal model was later included in KAOS to model security threats. The initial work also examined threats to online banking, but the scenario was limited to a single threat involving the compromise of account numbers and PINs [15, 17].

Attack-defense trees [9] are a recent parallel development to the KAOS framework. These trees extend attack trees by allowing nodes representing defensive measures to appear within a tree. Attack-defense trees are supported by a detailed theoretical model, but they do not have the tool support offered by KAOS.

Edge, *al.* [7] have employed a protection tree in an investigation of an online banking system to defeat various fraudulent methods modeled in the corresponding attack tree. The approach is extended in this paper by placing fraud methods and the corresponding protection measures in the same tree to clarify the relationship between fraud methods and fraud protection.

2.1 Goal Trees

The specification language of the KAOS framework has four domains: goal, operation, object and responsibility. This paper analyzes credit

card fraud in the goal domain. However, the other domains are also relevant to fraud analysis and will be the subject of future research.

A goal is an objective that a system is designed to achieve. An AND-refinement decomposes or refines a goal into a set of subgoals such that the satisfaction of all the elementary subgoals in the refinement is a sufficient condition for satisfying the composite goal. An OR-refinement relates a goal to an alternative set of subgoals such that that the satisfaction of one of the refined goals is a sufficient condition for satisfying the overall goal. Goal decomposition terminates when atomic goals called requirements are reached that can be directly executed (or "operationalized" in the KAOS terminology) by individual agents.

An obstacle [11] is a dual notion to a goal; it captures the possible undesirable conditions that frustrate a goal. Obstacles are a fundamental aspect of goal trees that facilitate detailed and practical analyses of how system goals may be breached. Obstacles can also be decomposed into finer and finer obstacles until they can be directly implemented at the level of anti-requirements, just like positive goal requirements. Finally, the resolution stage provides ways to counter the discovered obstacles so that the overall goals are satisfied even if undesirable issues occur.

An attack tree [13], like a goal tree, is also an AND-OR tree, except that an attack tree examines a system from an adversarial perspective instead of a defensive perspective. Goal trees are more functional and systematic than attack trees because the concept of obstacle is included directly with a tree along with the explicit linkage to the object, operation and responsibility domains.

Obstacle trees are sufficient for modeling and resolving inherent and inadvertent problems, but they are too limited for modeling and resolving malicious interference. The goal-oriented framework for generating and resolving obstacles was extended to address malicious obstacles called anti-goals [17], which could be executed by attackers to defeat security objectives.

3. Credit Card Transactions

A merchant's primary goal is to receive payment for the goods that are supplied. A scenario involving a remote payment is more difficult than when a customer purchases goods in person. This is because a credit card transaction relies on other system participants such as the card issuer, cardholder and courier to act correctly, and the evidence that is relied upon is often weak and open to challenge.

A merchant who accepts credit cards is committed to the rules of the card issuer such as Visa or MasterCard. If the transaction goes wrong,

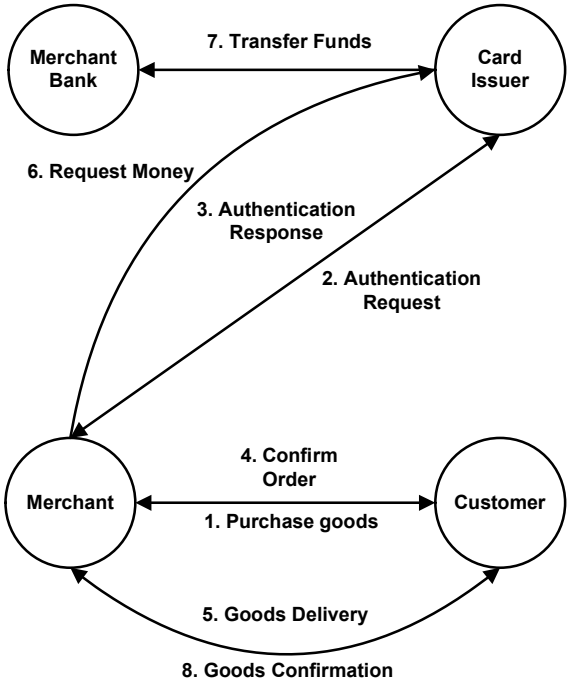

Figure 1. Card-not-present transaction.

the customer may receive a chargeback, which leads to the reversal of the payment. This situation can result in a loss for the merchant if weak authorization is used (e.g., provision of the card details alone), especially if the merchant cannot prove who initiated the transaction.

Internet credit card payments have complex security issues because the customer and merchant never meet and they rely on evidence communicated through potentially insecure channels using weak authentication measures that may be exploited by fraudsters. A remote credit card transaction over the Internet (using email or a website), or by phone, fax or regular mail is known as a "card-not-present" (CNP) transaction [14].

The EMV specification [8, 12] formally describes the process. The main parties to a payment card transaction are the merchant, the merchant's bank, the customer and the card issuer. There may be other participants, including a payment processor who performs transactions on behalf of the card issuer or merchant, and a courier who delivers the goods.

Figure 1 shows the EMV specification of a CNP transaction as a protocol exchange between the four main participants. The protocol involves several related flows of goods, information and money from one

participant to another in a temporal order, which is modeled later in this paper using goal trees.

A crucial issue is that the merchant may act on incomplete or incorrect information, because he/she may not be notified of fraud-related events if they occur (e.g., credit card theft or forgery). However, the merchant may be able to avoid liability for fraudulent transactions even with inadequate knowledge by passing the responsibility for fraud detection to another participant such as the cardholder or card issuer. In addition, the merchant can endeavor to collect sufficient auditing information to avoid liability when the merchant is responsible in the credit card system.

4. Fraud Analysis

Analysis of around 150 cases of fraud targeting banking systems has revealed that defective procedural controls and implementation flaws were exploited in nearly all the cases; only two cases involved the exploitation of weaknesses in cryptographic protocols, which received the most attention during system design [1]. A pragmatic and detailed fraud model can help merchants avoid or mitigate threats by imposing adequate obstacles.

The construction of a fraud tree involves building a KAOS goal tree from the attacker's perspective as in the case of an attack tree. It is useful to incorporate the attacker's perspective because the attacker's goals, motivation and activities could be missed when the focus is only on system goals.

In KAOS, the main obstacle corresponds to the logical negation of the goal that it is intended to defeat. An attacker goal that cannot be satisfactorily overcome indicates a failure of requirements engineering and the need to restructure the goal model. However, the credit card system is already in operation, so the obstacles under the attack goal may only be partially effective. In addition, some obstacles, such as the determination of card theft by the merchant, may be impossible to implement directly. We call these abstract obstacles, and they are forwarded to the later stages of a transaction for resolution.

Obstacles may be imperfect and incomplete, and can be overcome by further adversarial counter-goals unless additional obstacles are proposed. This is still useful because the merchant can take on transactions that might otherwise be rejected (an imperfect obstacle may be effective in the particular transaction context). In another context where the customer's identity cannot be established adequately, the definitive obstacle is to abandon the transaction after all the attempted checks fail.

4.1 Transaction Modeling

Building a fraud tree analyzes the threats to CNP from the fraudster's perspective. This is easier than decomposing the merchant goals and ensures that all plausible threats are recognized and addressed. It also provides an effective counterbalance against the idealized threat models that are produced when the focus is on the merchant's goals.

The model progressively decomposes the fraudster's goals into actionable steps as in the case of an attack tree. However, a fraud tree also contains defensive goals in the form of obstacles that can potentially defeat the adversarial goals. Because the defensive obstacles possibly offer imperfect and incomplete remedies, the process iterates through the fraudster's additional counter-goals and defensive obstacles for the counter-goals.

In requirements engineering, system threats are typically analyzed from all the stakeholders' points of view in order to formulate a collective system goal model. However, the participants in a transaction have their own goals, do not have complete visibility or control of the entire transaction system and may potentially be in an adversarial situation because a legitimate participant has to bear the cost of fraud.

At this stage, the fraud tree represents both perspectives – the merchant's and fraudster's – and must be transformed into a merchant-only view before use. Transforming the fraud tree converts abstract obstacles against adversarial activities that are invisible to the merchant to realizable obstacles that obstruct fraud in a different way. For example, although the merchant cannot detect the initial card compromise, the authenticity of the transaction can be confirmed when the goods are delivered by changing the payment to a local card-present transaction or by verifying the identity of the customer.

The transformation process begins with the initial fraud tree and imposes obstacles under each adversarial goal. The obstacle is purely abstract if it cannot be implemented, partial if it can be realized successfully under certain conditions, or total if it provides effective mitigation. In Figure 2, a total obstacle is represented using a rounded white square underneath the obstacle indicating success. Forwarded abstract obstacles (gray) and partial obstacles (lighter gray) extend into later steps in the transaction together with an annotation that indicates the circumstances causing the unresolved issue. All transaction flows should ideally end with resolved obstacles (white), but some light gray obstacles remain, indicating that, although fraud is significantly reduced, it is still possible.

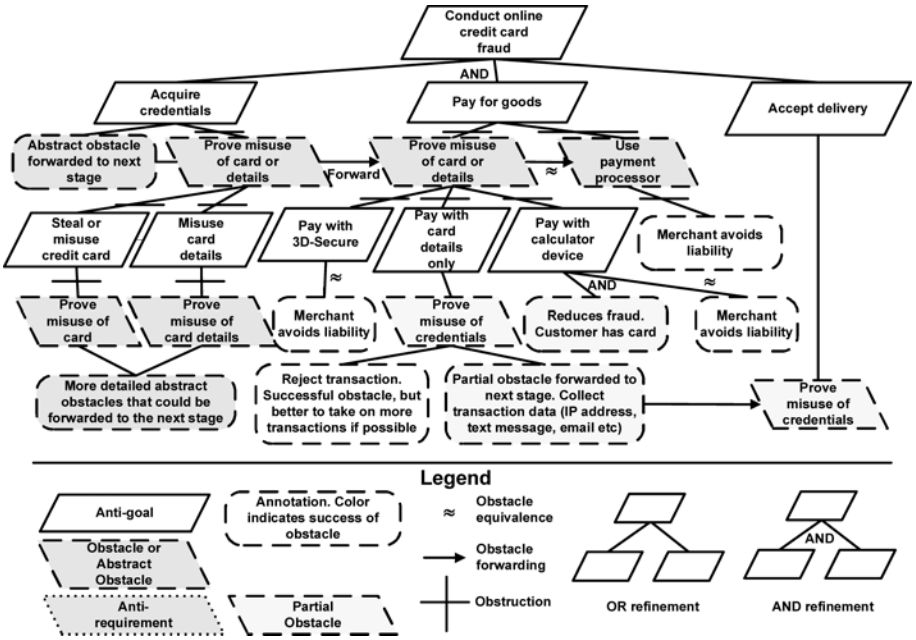

Figure 2. Fraud tree for the first two fraud stages.

Forwarded obstacles may be implemented directly to avoid fraud or they may be transformed to equivalent or weaker obstacles in order to avoid liability if fraud does occur. Realizable forwarded obstacles include avoiding fraud by aborting a suspicious transaction (Figure 2), reducing the probability of fraud by making additional identity checks during the delivery stage, avoiding liability by using the 3D-Secure payment mechanism (Figure 3) and collecting additional evidence to transfer accountability to the responsible party during the purchase and delivery stages.

The analysis is limited to the unauthorized use of a credit card to purchase tangible goods on the Internet as shown in the fraud tree in Figure 2. CNP fraud is the most common type of fraud. It is also the most challenging because the merchant does not see the card but, instead, uses a password for authentication. Card security codes along with PIN numbers can also be considered to be weak passwords, along with passwords used with 3D-Secure and for accessing merchant sites.

The obstacles for the first stage are negations of the fraudulent goals, which cannot be directly implemented by the merchant; thus, they are abstract obstacles that are forwarded to the second stage. A forwarded abstract or partial obstacle has an arrow between the parent and child that is annotated with the conditions for successful resolution. Because

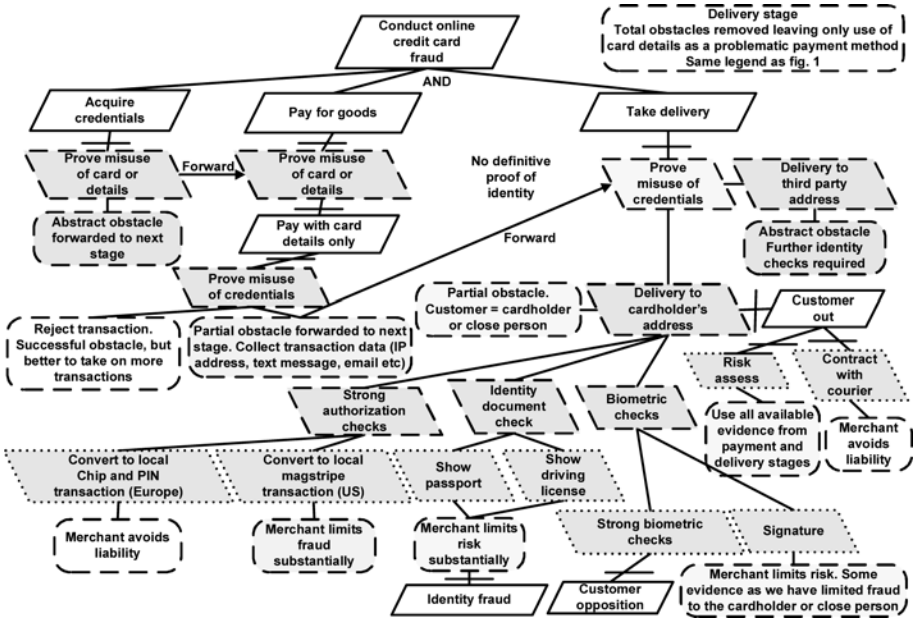

Figure 3. Delivery stage fraud tree.

the obstacle for the first stage is abstract, there is no annotation as no limitations are imposed on the transaction.

4.2 Fraud Tree

The resulting fraud tree shown in Figure 2 has three branches from the root AND-node that represent the three essential stages of fraud: (i) acquiring the card or card details; (ii) using them to purchase goods; and (iii) accepting delivery of the goods. Figure 2 shows the first two stages and Figure 3 presents the final delivery stage with the only unobstructed fraud path from the first two stages. The child nodes representing the three stages are subdivided into branches, recursively, until the decomposition terminates at unexpanded attack steps in the leaves that can be directly executed or that are deemed outside the scope of analysis.

The two possibilities for the first branch are to obtain the card or just the card details, which are equivalent in online transactions because the card is never seen and is not directly used for authorization, except for the relatively rare case when a calculator-like device is used to compute a one-time password for each transaction. However, having the card or just the card details can be distinguished by later checks, so they can be forwarded as different obstacles to the delivery stage. It is far easier for a fraudster to discover card details than to acquire a card because the card

details are provided to a merchant or proxy (e.g., payment processor) in every transaction.

Card details can be compromised in multiple ways, but this analysis limits it to someone close to the cardholder or an unknown third party. Finally, the transaction could be carried out and subsequently denied by the cardholder, which is impossible to demonstrate under the current transaction system, and is a reason why additional forensic evidence should be collected to establish the identity of the customer.

The second branch is to select and pay for the goods, where the different payment methods are the different anti-goals that can be satisfied by the adversary. The obstacle for "Acquire credentials" is also the child of the "Pay for goods" node in the second branch.

3D-Secure is a good payment method when the cardholder is deemed responsible or negligent for fraud (e.g., for revealing the card details and password). The card issuer typically provides the purchaser with a popup window in which a password is entered [10]. The obstacle for avoiding fraud forwarded from the first stage is transformed from discovering the identity of the customer to the alternative acceptable obstacle of avoiding liability (shown by the white annotation indicating success and the $\approx$ symbol for transformation to an equivalent obstacle).

Another possibility is to use a calculator-like device to compute a one-time password that restricts fraud to the less common case of having access to the card. This could have been shown as a total obstacle for the compromise of card details (stops the fraud) and partial obstacle for card theft (fraud is still possible) if both classes had been forwarded separately. However, by forwarding the two fraud methods as a single class, the measure is only a partial obstacle for the entire class (shown as light gray in Figure 2). The method also avoids merchant liability because the card issuer assumes the responsibility, indicated by the satisfied obstacle under the calculator payment node (colored white).

For the situation where only the card details are supplied, a complete obstacle is to reject any weakly authorized transaction using only the card details and card security code. However, merchants often allow weak authentication to take on more business, because their goal is to increase profits instead of avoiding fraud entirely.

An additional way to resolve the obstacle is to engage a payment processor who decides on the legitimacy of transactions; this transfers merchant liability for a fee. The transformation of the forwarded obstacle "Prove misuse of card or details" to "Use payment processor" for a fixed cost is usually a good option because it also avoids administrative effort and further security issues outside the scope of the scenario (e.g., disclosure of sensitive cardholder details). Thus, the obstacle at the

root of the payment branch is resolved if the merchant chooses to use a payment processor.

The outcome of the first two transaction stages leads to one unsatisfied obstacle that is forwarded to the delivery stage. Figure 3 presents the fraud tree for the delivery stage. When the merchant insists on delivery to the cardholder's address, the analysis can extend to establishing additional checks that the merchant can perform when the goods are delivered elsewhere.

Fraud in the case of home delivery is only possible when the customer is the cardholder or is in close proximity to the cardholder and can take delivery of the goods. After the partial obstacle provided by the address check, the significant issue of remote fraud is avoided and the customer's identity is narrowed down to the cardholder or someone close.

It is necessary to collect further evidence to show that the cardholder is responsible because it is not adequate to claim that, since all the other possibilities are ruled out, the cardholder or someone close to the cardholder must have executed the transaction. For example, fraud is not ruled out on the part of the merchant, the courier and their employees.

Most fraud checks are ineffective against insiders, such as cardholders who act legitimately until they claim that they did not carry out certain transactions. Alternatively, it is entirely possible for a friend, colleague or family member to carry out fraud successfully without detection, so it is imperative that the merchant can establish attribution by collecting additional evidence outside the transaction system.

We summarize the situation if the third stage is reached without establishing the identity of the customer or avoiding liability. The first point is that it is not known if the transaction is legitimate or fraudulent; therefore, it is important not to execute clumsy and ill-directed checks when the vast majority of transactions are legitimate. A reasonable assumption is that the cardholder or someone close to the cardholder executed the transaction if the goods were delivered to the cardholder's address.

The merchant needs to augment the system when weak payment authentication is used by conducting additional verification checks to limit fraud or collect additional evidence to avoid liability. Note that, when a transaction is fraudulent, it is not known if the card or card details were misused. It is sensible to assume the worst that the card was stolen, but the two cases can be distinguished and different checks can be conducted to avoid them.

If the transaction circumstances suggest that a person close to the cardholder might have misused the card details, then it is reasonable to insist on a local card transaction. However, if the cardholder is im-

plicated, then it would be more reasonable to ask for stronger identity checks and to use both types of checks if the identity of the recipient is unclear. The different types of control systematically provide obstacles to each type of fraud proactively, before the fraud method is known and even before it is known that the fraud has occurred. The controls have to be lightweight enough not to discourage the vast majority of legitimate purchasers. Therefore, onerous verification checks should only be applied to high value or suspicious transactions.

The major issues with physical delivery are practical concerns such as the need to provide fallback checks if the safest methods are unavailable. These practical issues are often inadequately analyzed using attack trees and other approaches. A crucial issue, modeled by an additional adversarial goal that defeats the obstacle of requiring strong authentication, is that the customer may not be at home and a neighbor or someone else at the address accepts the goods.

The possibility of fraud cannot be eliminated easily, but liability can be avoided by specifying a contract with the courier that transfers the fraud detection responsibility to the courier. This passes the risk assessment decision to the courier who decides whether to deliver the goods to a third party or to return when the customer is at home. This transformed obstacle of avoiding liability is a sufficient obstacle for the merchant, who also avoids all the logistical and security issues regarding delivery while passing the risk assessment decision to the courier, to whom it most sensibly belongs.

5. Conclusions

The use of fraud trees to analyze Internet credit card fraud can systematically provide an obstacle to each type of fraud proactively before the fraud method is known. By including obstacle formation and transformation, fraud trees are more refined than attack trees and adopt a different perspective compared with goal trees. An interesting aspect is that *a priori* knowledge of the branches of the fraud tree occupied by a transaction is required because the tree includes countermeasures to deal with each type of fraud. Fraud trees have applications to other types of investigation where wrongdoing is discovered after the fact, as in the case of the insider threat. Many insider threat incidents cannot be stopped, but it is possible to collect sufficient evidence to hold the perpetrators responsible.

Other benefits of the fraud tree framework include completeness (all known fraud techniques can be analyzed), scope (while the focus is on the logical transaction, incorporating physical and social checks helps

reduce fraud and liability), participant perspective (participants do not share the same goals and can be in an adversarial position when there is a successful fraud, so it is useful to consider what each participant knows and can control), adversarial perspective, and narrative structure (the security measures used by the merchant are incorporated in the fraud tree and help explain the fraud).

Our future research will attempt to develop a firm theoretical foundation using temporal logic and model checking. Also, the use of binary yes/no measures is less than satisfactory; incorporating probabilistic measures of fraud and the costs of countermeasures will enhance risk assessment. Another related topic is to combine probabilities and other numerical measures as in the case of KAOS goal trees [6]. Other research topics involve the examination of partially satisfied obstacles that incorporate weaknesses that could be targeted by fraudsters, and the estimation of the intangible costs of performing checks.

References

[1] R. Anderson, Why cryptosystems fail, *Proceedings of the First ACM Conference on Computer and Communications Security*, pp. 215–227, 1993.

[2] B. Aziz, Towards goal-driven digital forensic investigations, *Proceedings of the Second International Conference on Cyber Crime, Security and Digital Forensics*, 2012.

[3] B. Aziz, C. Blackwell and S. Islam, A framework for digital forensics and investigations: The goal-driven approach, *International Journal of Digital Crime and Forensics*, vol. 5(2), pp. 1–22, 2013.

[4] C. Blackwell, B. Aziz and S. Islam, Using a goal-driven approach in the investigation of a questioned contract, in *Advances in Digital Forensics IX*, G. Peterson and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 153–167, 2013.

[5] M. Bond, O. Choudary, S. Murdoch, S. Skorobogatov and R. Anderson, Chip and skim: Cloning EMV cards with the pre-play attack (`arxiv.org/pdf/1209.2531.pdf`), 2012.

[6] A. Cailliau and A. van Lamsweerde, Assessing requirements-related risks through probabilistic goals and obstacles, *Requirements Engineering*, vol. 18(2), pp. 129–146, 2013.

[7] K. Edge, R. Raines, M. Grimaila, R. Baldwin, R. Bennington and C. Reuter, The use of attack and protection trees to analyze security for an online banking system, *Proceedings of the Fortieth Annual Hawaii International Conference on System Sciences*, p. 144b, 2007.

[8] EMVCo, EMV 4.3, Otley, United Kingdom (`www.emvco.com/specifications.aspx?id=223`), 2011.

[9] B. Kordy, S. Mauw, S. Radomirovic and P. Schweitzer, Attack-defense trees, *Journal of Logic and Computation*, 2012.

[10] S. Murdoch and R. Anderson, Verified by Visa and MasterCard SecureCode: Or, how not to design authentication, in *Financial Cryptography and Data Security*, Springer-Verlag, R. Sion (Ed.), Berlin Heidelberg, Germany, pp. 336–342, 2010.

[11] C. Potts, Using schematic scenarios to understand user needs, *Proceedings of the First Conference on Designing Interactive Systems: Processes, Practices, Methods and Techniques*, pp. 247–256, 1995.

[12] C. Radu, *Implementing Electronic Card Payment Systems*, Artech House, Norwood, Massachusetts, 2002.

[13] B. Schneier, Attack trees, *Dr. Dobbs Journal*, vol. 24(12), pp. 21–29, 1999.

[14] The U.K. Cards Association, Card-not-present transactions, London, United Kingdon (`www.theukcardsassociation.org.uk/cards-transactions/card-not-present.asp`).

[15] A. van Lamsweerde, Elaborating security requirements by construction of intentional anti-models, *Proceedings of the Twenty-Sixth International Conference on Software Engineering*, pp. 148–157, 2004.

[16] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*, Wiley, Chichester, United Kingdom, 2009.

[17] A. van Lamsweerde, S. Brohez, R. De Landtsheer and D. Janssens, From system goals to intruder anti-goals: Attack generation and resolution for security requirements engineering, *Proceedings of the Workshop on Requirements for High Assurance Systems*, pp. 49–56, 2003.