Clarke, Charles, Pfluegel, Eckhard and Tsaptsinos, Dimitris (2015) Multi-channel overlay protocols : implementing ad-hoc message authentication in social media platforms.In 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). Institute of Electrical and Electronics Engineers, Inc. ISBN 9780993233807.
http://dx.doi.org/10.1109/CyberSA.2015.7166118

Multi-channel Overlay Protocols: Implementing Ad-hoc Message Authentication in Social Media Platforms

Charles A. Clarke
Faculty of SEC
Kingston University
Penrhyn Road
Kingston upon Thames
Surrey, KT1 2EE
Email: k0840680@kingston.ac.uk

Eckhard Pfluegel
Faculty of SEC
Kingston University
Penrhyn Road
Kingston upon Thames
Surrey, KT1 2EE
Email: E.Pfluegel@kingston.ac.uk

Dimitris Tsaptsinos
Faculty of SEC
Kingston University
Penrhyn Road
Kingston upon Thames
Surrey, KT1 2EE
Email: D.Tsaptsinos@kingston.ac.uk

***Abstract*—As businesses, governments and professional institutions progressively seek to engage with consumers via social media platforms (SMPs), the capacity of SMP users to validate the source of received content and its integrity, becomes increasingly significant. Historically, SMPs have an associated legacy of security concerns, many of which pertain to content integrity. In this paper, we present designs for multi-channel overlay protocols, that are used to implement ad-hoc authentication of user-generated content (messages), in social media platforms. Our approach draws inspiration from protocols that are conventionally used for pairing wireless devices in ad-hoc networks. Hence, we compare and contrast conventional device pairing protocols with our own, as well as consider the security characteristics, benefits and limitations of our protocols.**

I. Introduction

The scope of *user-generated content* (messages) shared via *social media platforms* (SMPs) is as vast as it is diverse. User-generated content is typically reflective of the multifarious social interactions, that occur between friends and special interest groups. Increasingly however, interactions in SMPs (examples of which include Facebook, Line, LinkedIn, Kik, SnapChat, Twitter, WhatsApp, and YouTube) are used as data sources for Social *Customer Relationship Management* (socialCRM) tools , described in [1] as an emerging sector, that enables businesses, organisations and government institutions to dynamically engage with their consumers in a way, that is beyond conventional channels (e.g. email, surveys etc). However, as complex multi-user applications, SMPs have spawned a legacy of security concerns, many of which pertain to the *confidentiality*, *integrity* and *availability* (CIA) of user-generated content.

In our previous research published in [2], we described how SMPs posed specific confidentiality threats to user-generated content and proposed the use of *virtual private social networking* techniques (a concept proposed in [3]) to mitigate such threats. We also specified how content integrity can be detrimentally impacted by format, capacity and semantic constraints that are routinely imposed by some SMPs, thus deeming them to be *untrusted*.

A. Research Contribution

In this paper, we are primarily concerned with mechanisms for authenticating the integrity of content that is shared between SMP users. Therefore, we conceptualise and propose designs for *multi-channel overlay protocols* that can be applied by SMP users in an ad-hoc manner. Verifying content integrity, is relevant to scenarios that utilise time stamped legal documents, files that are analysed under forensic investigations and journalistic content, where statements, images and videos may need to be validated as unaltered and genuine. The motivations for our approach are based on concepts derived from hash based protocols that are conventionally applied to pairing wireless devices in ad-hoc networks. As a hash based family of device pairing protocols, we note that concepts published in [4], [5], [6] and [7] are particularly relevant to our work and therefore, we collectively refer to them as *hash based device pairing* protocols, in this paper.

B. How this paper is organised

In Section II, we present related work in the context of motivations for our approach, as well as requisite terminology and concepts that are fundamental to this research. We present our protocol concepts, techniques and security evaluations, in Section III and the benefits and limitations of our protocols are outlined in Section IV.

II. RELATED WORK

As a background to our research, we refer to hash based device pairing protocols, specifically research published in [6] and MANA protocols presented in [7]. Therefore, in this section we define requisite terminology that will be used throughout this paper and outline hash based device pairing protocol concepts and implementation assumptions. This section concludes with a definition of some notations and an outline of protocol event sequences.

A. Requisite Terminology

We use the term *entity* to refer to a device, user or host within a network. Entities interact with each other as *senders*

and *recipients* (in the context of message integrity) and as *claimants* and *verifiers* (in the context of origin integrity). The term *channel* is used to describe a communication link that is established between entities. These links may be physical or logical, secure or insecure, and wireless or wired. Channels may also have specific security characteristics that classify them as *Out-of-Band* (OOB), thus deeming them suitable for secure and private message transmissions. A *message* (i.e. data), is defined as the means by which entities interact via channels (i.e. entities exchange messages). A message can take many forms including binary, hex, text, image, audio and video.

1) Message and Origin Authentication: The goal of establishing message integrity (message authentication) aims to ensure that a message shared between a sender and a recipient is preserved and validated to be exactly as issued by the sender (i.e. a digest of a message calculated at source and destination, would be identical). Establishing origin integrity (origin authentication) aims to validate the source of where a message is believed to have originated from, such that a message sent to a verifier called *Bob*, that is believed to have originated from a claimant called *Alice*, can be trusted to have actually originated from the claimant called Alice (i.e. it was not intercepted, modified or re-transmitted by a malicious third party).

2) Authenticated Channels: To contextualise additional protocol security considerations, we refer to work published in [8], in which the author considers the security of channels in terms of *weak authentication* and *strong authentication*. A weak authenticated channel (also referred to as an *authenticated channel*) denotes that messages transmitted via a given channel, cannot be changed. However, the transmission of a message may be subject to attacks in which they may be malevolently interrupted, removed, or replayed. The author informally describes a strong authenticated channel as an authenticated channel that possesses an additional security characteristic, such that one or more of these attacks can be mitigated. Examples of such characteristics include *stall-free* transmission of a message, *acknowledgement* by a verifier to a claimant of receiving a message and *verifier-ready* transmissions, that allow a claimant to check that a verifier is awaiting incoming messages on an authenticated channel. Based on these security properties, a human OOB channel can be described as a strong authenticated channel in the context of the hash based device pairing protocols that we reference.

B. Device Pairing Concepts and Implementation Assumptions

In [6], [7] and [9] (the underlying hash based device pairing protocols considered in this research) a common scenario is presented in which two wireless devices (e.g mobile phones, keyboards, speakers, displays etc) want to identify and securely pair with each other over a wireless link. This pairing step (referred to in [6] as a *pre-authentication* step), is typically instigated prior to the devices implementing further protocols (e.g. key agreement, channel encyption etc). These protocols require two channels, hence the proposal to formally acknowledging them as *multi-channel* protocols proposed by the authors in [9]. The first channel is wireless and is considered to be *untrusted*. The second channel is designated as a secure OOB channel and is considered to be *trusted*. Note that the OOB channel is typically a 'human' channel, in that it relies on human intervention, scrutiny and actions to achieve strong authenticated security characteristics.

These non-interactive protocols (i.e. a protocol that requires transmission activity from only one party) have two fundamental stages. In the first stage, the claimant sends a secret key and message verification via the OOB channel to a verifier. The message verification may typically be derived from a cryptographic primitive, such as a secure hash function, hashed based message authentication code (HMAC) or commitment scheme. Such that the message authentication is short enough to be of practical use when being manually applied by human actions, and long enough to provide a secure key space. In the next stage, the claimant sends a message via the wireless channel to the verifier, who uses the secret key and message verification (previously received via the OOB channel) to authenticate the message received via the wireless channel. If authentication is successful, the verifier can trust that the originator of both the message, message verification and key, have all originated from the same source, thus enabling further protocol implementations to commence (e.g. channel encryption, key exchange etc). These protocols are implemented based on the assumptions described in II-C.

C. Device Pairing Protocol Assumptions

The communication channel between devices is typically wireless and is deemed to be insecure, where messages transmitted may be subject to threats associated with man-in-the-middle attacks, namely eavesdropping, interception, interruption, modification and replaying.

The devices may not necessarily have matched input and output capabilities (e.g. a mobile phone may have a screen and keyboard, but wireless speakers might only have an LED and function buttons).

The devices cannot use a public key infrastructure or any trusted third party certification authority to aid pairing.

The devices can communicate via an additional channel (OOB) that is assumed to be controlled by a single human or pair of users who trust each other. Integrity of this channel is assumed to be established and requires human judgement and actions (e.g. entering a pin or capturing a token via a camera, touching devices together etc) to function. This effectively obtains a strong authenticated channel as per the concepts described in Section II-A2.

D. Notations and Protocol Event Sequences

We denote A as a claimant device that must be authenticated, and B as a verifier device that will validate A. Let w be a wireless channel and o, an out-of-band channel. Finally, let m denote a message, k a random secret key, h a cryptographic hash function and d a message digest. The protocol event sequences are outlined as follows:

A **calculates** k
A **calculates** d = $h(k||h(m))$
A **transmits** k and d over o to B
A **transmits** m over w to B
B **calculates** $\hat{d}$ = $h(\hat{k}||h(\hat{m}))$
B **compares** $\hat{d}$ and d

A generic diagram depicting hash based device pairing protocols is shown in Figure 1.

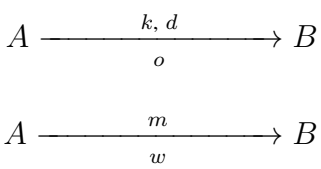

Fig. 1. Generic hash based device pairing protocol sequence

A match between $\hat{d}$ and d, authenticates A. Having successfully authenticated A, conventional channel encryption or key exchange protocols etc, can be implemented. As shown in [7], this protocol achieves security against a man-in-the-middle attack, based on the assumptions described in Section II-B and the premise that an adversary has the scope to control the wireless channel but not the human OOB channel. However, we note that in [9], the authors suggest that this assumption can no longer be made for human OOB channels, due to the pervasiveness of CCTV.

III. OUR APPROACH

We recall from the introduction to this paper, that our research is concerned with multi-channel overlay protocols and how they can be used to establish ad-hoc authentication of user-generated content, shared between SMP users. We note that not all content shared between users will need to be authenticated, thus we use the term ad-hoc in the context of protocols that can be applied over a dynamic choice of channels, as and when they are required. In this section, we present requisite terminology and describe our approaches for two specific multi-channel protocols. For each protocol, we describe our approach, define some notations, outline protocol event sequences, consider the impact of channel eavesdropping and modification on each protocol and summarise implementation requirements. The benefits and limitations of each protocol are considered in section IV.

A. Terminology and Concepts

We begin by mapping definitions for the terms 'entity', 'channel' and 'message' (previously described in Section II-A) to equivalent objects in the context of SMPs. Therefore, SMP user accounts are considered to be analogous to entities, SMPs to channels and user-generated content to messages. We note that entities may share messages *intrinsically*, via a channels publishing infrastructure (e.g. page, wall, post, tweet, alert etc) or *extrinsically*, via a channels private messaging or file attachment features, where a recipient entity's intention is to access a message outside of the channel through external tools (e.g. image reader, media player, decompression software).

1) Channel Threats to Message Integrity: SMP channels are unauthenticated according to the definition proposed in [8] and as articulated in Section II-A2, however they are distinct and autonomous hosts, that impose security mechanisms that aim to protect their systems and users. As publically accessible applications, it is in the interests of many SMPs to proactively maintain and improve security. Therefore, we assume for example that SMPs establish encrypted links between remote users and themselves (e.g by implementing https sessions) and utilise challenge-response password mechanisms etc. To this end, we limit our security concerns to those that might occur within the confines of the SMP.

Theoretically (and practically), some SMP channels have the scope to conduct the kind of malicious activities that are conventionally associated with a *man-in-the-middle* (MitM) attack, namely *eavesdropping*, *interception*, *interruption*, *modification* and *replaying* messages. However, whilst the practice of interception, interruption and replay actions are clearly not in the best interests of a channel's business model, entities have no mechanism for verifying that eavesdropping does not take place or restricting SMP applied modifications to messages, that may be actioned for functional reasons. For example, a channel could legitimately access and analyse messages for keyword searches and behaviour tracking purposes, and deem the practice necessary in the context of revenues derived from targeted advertising. It would seem clear that eavesdropping in this context is not intentionally malicious, but may still be perceived by some entities, as a threat. Likewise, modification of a message by channels for the purposes of conforming it to a pre-designated format or specification (e.g. format conversion, cropping or resizing an image, modifications to meta-data etc) may not be deliberately malicious, but may be perceived as a threat to content integrity none the less.

A further integrity concern, results from the requirement to publish messages that are contextually relevant (i.e. appropriate messages in an appropriate context). For example, contextual relevance limits the use of publishing encrypted messages as they would appear out of place and potentially raise the suspicions of the channel, perhaps resulting in messages being sanitised or an account entity being suspended. Additionally, the imposition of a capacity limit on messages may also impact message integrity. For example, Twitter was originally designed to be text-centric and imposes a 140 character limit on *tweets*, thus the original form of a message may have to be modified, in order to comply with specified capacity limits. In short, messages that are shared via channels, must be transmitted in a manner that mitigates any integrity threats posed by some channels. To this end, SMP's that are used in multi-channel protocols, cannot be designated as trusted out-of-band channels in accordance with the notions described in [5], [7] and [6]. Therefore, their practical use in multi-channel protocols, relies on certain assumptions that are detailed in Section III-B.

B. SMP Multi-channel Overlay Protocol Assumptions

SMPs are trusted to maintain origin authentication mechanisms between entities (as described in Section II-A1). Origin integrity is a fundamental requirement for a reliable, practical and convenient SMP, therefore, we consider this to be a fair assumption.

We assume that a single adversary may control no more than a single channel and that channels do not collude or have an implied collusion. For example, Facebook own WhatsApp and Instagram, therefore these channels would not be paired in a multi-channel protocol.

It is assumed that communication links between an entity and SMPs are https secured, confining man-in-the-middle based attacks to within the SMP infrastructure. However, whilst we consider interception, interruption and replay attacks by SMPs on it's users to be contradictory to business interests and therefore highly unlikely, we accept that eavesdropping and modification threats remain.

In the context of legitimate use of these protocols within SMPs, we assume that detection by the SMP of authentication activity is not considered to be malicious, unless it somehow impairs their business model or contravenes their terms and conditions of use.

We assume that communicating entities have appropriate and fully functioning accounts on relevant SMPs and that they trust each other.

The entities do not use a public key infrastructure or any trusted third party certification authority to validate message integrity.

C. Protocol 1

The approach in this protocol, generally mirrors that of the hash based device pairing protocols described in Section II-B, with some notable differences. Firstly, Protocol 1 is not constrained to using digests of a restricted size (e.g. a digest does not have to be truncated as it is not processed manually by 'hand'). Secondly, SMP channels are unauthenticated and therefore cannot be utilised as OOB channels, which by definition are strong authenticated channels. Consequently, in Protocol 1, channels are limited to a usage scenario in which confidentiality of messages cannot be assumed, thus rendering the use of a secret key as irrelevant.

Like the hash based device pairing protocols, Protocol 1 has two fundamental stages. In the first stage, the message sender transmits a message verification via an elected SMP channel to the recipient. In the second stage, the sender transmits a message to be authenticated, via the alternate SMP channel, to the recipient. The recipient uses the message verification previously received via the first SMP channel, to authenticate the message received via the second SMP channel. Message authentication is successful if the recipient can reproduce the message verification value.

1) Notations and Protocol Event Sequences: We denote A as a *sender* entity that transmits a message m to a *recipient* entity, B. Let s_0 and s_1 be two separate and non-colluding SMP channels, h a cryptographic hash function and d a message digest. The sequence for this protocol is shown in Figure 2 and is expressed as follows:

A **calculates** $d = h(m)$
A **transmits** d over s_0 to B
A **transmits** m over s_1 to B
B **calculates** $\hat{d} = h(\hat{m})$
B **compares** $\hat{d}$ and d

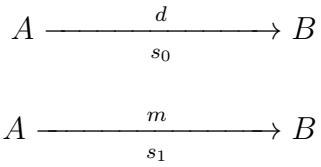

Fig. 2. Protocol 1 - Event Sequence

Theorem 1. *If $\hat{d} = d$, the integrity of m is verified hence $\hat{m} = m$. If $\hat{d} \neq d$, the integrity of m is not verified and B only knows that either d or m have been modified.*

Proof: Modification to d by channel s_0 would mean that B cannot not authenticate m. Likewise, if channel s_1 modifies m in some way, m cannot be authenticated by B. Therefore, B can authenticate m under the assumption that neither channel applies modifications, but in the event that authentication does fail, B cannot determine which channel applied the modifications. ■

2) The Implications of Channel Eavesdropping: Entity B wants to authenticate the integrity of m received from entity A, via channel s_1. To achieve this, entity B requires d that is received via channel s_0. Channel s_0 eavesdropping on d, cannot associate this content with m on channel s_1, therefore channel s_0 has access to a digest, but nothing that it pertains to. However, if d is not published in a contextually relevant manner, it may appear as suspicious to channel s_0. To mitigate this problem, A must ensure that d is published in a contextually relevant manner. Assuming that m is also contextually relevant, eavesdropping by channel s_1 would reveal generic and innocuous content that does not appear suspicious. Thus we conclude that providing d and m are published in a way that is contextually relevant and does not contravene the terms and conditions of the SMP, that eavesdropping by a channel does not impair the functionality of this protocol.

3) Implementation: Participants require access to a cryptographic hash function tool. They should also have user accounts on multiple SMPs and must pre-agree a selection of SMPs that may be used as channels. Recipient B, also needs to be able to identify verifiable content. B can achieve this by accepting that the receipt of d over s_0, indicates that m, sent via s_1 can be authenticated. An additional recipient acknowledgement step can be added, requiring B to return an agreed acknowledgement message to A. For example, this could be a text-based response message that features a particular emoticon sent via channel s_0. Adding this step would transform this protocol from being non-interactive, to interactive as well as adding a strong authenticated characteristic to channel s_0, although the channel would still be considered as unauthenticated.

D. Protocol 2

The concept for our second approach is some what unconventional and can be viewed as a commitment scheme. It yields notable security properties that are evaluated in Section IV. This protocol requires the use of two SMP channels and can be implemented non-interactively or interactively, dependent on the participants requirements. It also relies on the use of an additional SMP account (i.e. one that has been previously created by the sender) that is used exclusively for the purpose of authenticating transactions; thus we will refer to this account as an *authentication transaction account* (ATA). Details of the ATA (i.e. SMP and login name) must be pre-shared with intended recipients.

The stages of this protocol are defined as follows. The sender creates a message (e.g. an image) and calculates a message digest. The sender logs in to the ATA and sets the ATA password to the value of the message digest. The last action on the sender's part, is to transmit the message to the recipient, via the alternate SMP (i.e. the SMP on which the ATA does NOT reside). On receipt of the message, the recipient calculates the digest of the message, and uses it as a password to login to the ATA. A failed login would indicate that the message received had been modified, whilst a successful login indicates message integrity. This protocol becomes interactive, if the recipient sends an acknowledgement message to the sender whilst logged in to the ATA. In order for this protocol to be implemented, the following additional assumptions to those described in Section III-B, have to be made:

A sender must create and maintain an ATA, which must be created prior to the implementation of the protocol.

Appropriate details of the ATA (i.e. SMP name, login name) must be pre-shared with trusted recipients.

The ATA password mechanism must have the capacity to store a password that is equal to the length of a message digest.

1) Notations and Protocol Event Sequences: We denote A as a *sender* entity, B as a *recipient* entity, T as an ATA entity and r as a password. Let s_0 and s_1 be two separate and non-colluding SMP channels, m a message, m_a an acknowledgement message, h a cryptographic hash function and d a message digest. The sequence for this protocol is shown in Figure 3 and expressed as follows:

A **calculates** $d = h(m)$
A **logs in to** T over s_0 and sets $r = d$
A **transmits** m over s_1 to B
B **calculates** $\hat{d} = h(\hat{m})$
B **logs in to** T over s_0 using $\hat{d}$ as r
B **transmits** m_a over s_0 to A as T

A successful login to T by B using $\hat{d}$ as a password, indicates message integrity, whilst a failure to login would indicate modification to m.

2) The Implications of Channel Eavesdropping and Modifications: Eavesdropping by channel s_0 might observe a change of r for T. We note that eavesdropping may also reveal user activity tracking and IP address data that could be utilised by s_0 to monitor the account activity of T. However, a password reset should not in itself be deemed as suspicious and therefore contextual relevance is not an issue, as it could be in Protocol 1. It is assumed that m is published in a contextually relevant manner, and thus eavesdropping by channel s_1 would likewise reveal innocuous and unremarkable content.

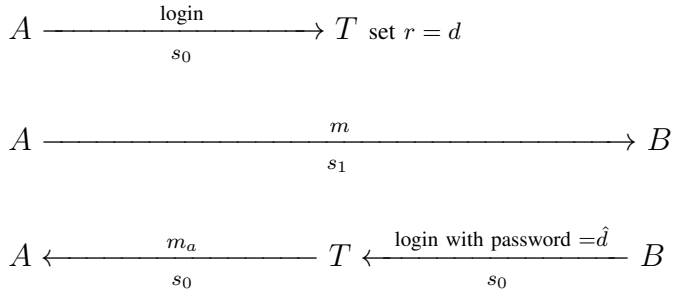

Fig. 3. Protocol 2 - Event Sequences

Modification of r by s_0, is highly unlikely as it would be against the business interests of SMPs to modify user passwords. Therefore, integrity of r may be deemed to be somewhat resilient. Like Protocol 1 however, channel s_1 could modify m in some way, meaning that B would be unable to authenticate m. Therefore, B can authenticate m under the assumption that channel s_1 does not apply modifications and in the event that authentication does fail, B can assume that the modification was applied by channel s_1.

3) Implementation: The participants of this protocol require access to a cryptographic hash function tool and like Protocol 1, they should have user accounts on multiple SMPs and pre-agree a selection of SMPs that may be used as channels. Additionally, this protocol requires a sender to create and maintain a dedicated ATA account for each SMP channel, the details of which (i.e. SMP, login name) must be shared with trusted recipients. Unlike Protocol 1, this protocol has no intuitive way to indicate that m can be validated. Therefore, we propose a simple example based on the use of emoticons, such that when A sends m to B, that m features an agreed emoticon signifying that m can be authenticated. This protocol can be described as an interactive protocol, that includes a strong authenticated feature on channel s_0, as described in Section II-A2. However, like Protocol 1, the channels are still considered unauthenticated.

IV. Conclusions

In this paper, we have proposed and evaluated two multi-channel overlay protocols that enable users in social media platforms (SMPs), to apply ad-hoc authentication of user-generated content shared in SMPs. Our approach has been influenced by protocols that are conventionally used for pairing wireless devices in ad-hoc networks, where they are utilised to mitigate threats associated with *man-in-the-middle* attacks. Therefore, in this section, we clarify differences in protocol goals and characteristics, that demarcate this research from the conventional use of protocols in device pairing and conclude with a summary of the benefits and limitations for the protocols we have proposed.

A. Hash based device pairing protocols and SMP multi-channel overlay protocol disparities

The first disparity between the protocols lies in their goals, where device pairing focuses on ad-hoc entity authentication and our protocols focus on ad-hoc message authentication. Similarly, the assumptions between channel properties also differ. In device pairing the main channel is assumed to be wireless and untrusted, and an additional trusted OOB channel, is a requisite factor in order for the protocol to function. In our approach, both channels are SMPs and untrusted as they do not poses the security characteristics of a human OOB channel as described in [6] and [7]. Furthermore, in our protocols the link between an entity and the SMP is assumed to be secured using https and we assume that entity authentication is imposed by the SMP (as per our assumptions detailed in Section III-B.

Another major disparity relates to the origin of security concerns. The security concerns for hash based device pairing protocols pertain to man-in-the-middle related threats over the wireless channel. These threats are mitigated based on the assumption that messages are sent via two channels and that all messages must be accessed in an unaltered manner, to authenticate an entity. The security of this approach relies on the notion that an attacker cannot access all channels (one of which is an OOB channel). In our overlay protocols, we rely on the same notion for security, however our security concerns are considered to originate from the SMP rather than conventional man-in-the-middle attacks.

B. Benefits and Limitations

Protocol 1 has the benefit of being easy and intuitive to implement as well as having a scalability advantage over Protocol 2. However, it has the limitation that message authentications (e.g. digests, commitments, HMACs) must be published in a manner that is contextually relevant. In some SMPs this may be problematic, however can be mitigated by concealing message authentications as payloads in steganographic carriers.

Protocol 2 is somewhat unconventional in that is relies on the use of an account password to securely share a message authentication with a recipient. It's benefits include the sharing of a message authentication in a manner that precludes it from modification (a characteristic that does not feature as part of the first protocol.) We note that use of an ATA in this protocol, may contravene the terms and conditions of some SMPs. Furthermore, frequent changing of an account password, might be deemed as suspicious behaviour by some SMPs. We also note that the account registration processes of some SMPs can be extensive, sometimes requiring an email, mobile number and private address details. Additionally, some password update mechanisms require email validation. This combination of account creation and password updates, add management and administrative overheads that may restrict the use of this protocol to very small user groups.

C. Further Research

It is envisaged that further research, will combine techniques described in this paper with steganographic techniques, thus achieving security characteristics that establish message confidentiality, contextual relevance and message authentication.

REFERENCES

[1] C. Heller Baird and G. Parasnis, "From social media to social customer relationship management," *Strategy & Leadership*, vol. 39, no. 5, pp. 30–37, 2011.

[2] C. Clarke, E. Pfluegel, and D. Tsaptsinos, "Confidential communication techniques for virtual private social networks," in *Distributed Computing and Applications to Business, Engineering & Science (DCABES), 2013 12th International Symposium on*. IEEE, 2013, pp. 212–216.

[3] M. Conti, A. Hasani, and B. Crispo, "Virtual private social networks," in *Proceedings of the first ACM conference on Data and application security and privacy*. ACM, 2011, pp. 39–50.

[4] D. P. Maher, "Secure communication method and apparatus," Sep. 12 1995, uS Patent 5,450,493.

[5] F. Stajano, "The resurrecting duckling," in *Security Protocols*. Springer, 2000, pp. 183–194.

[6] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks." in *NDSS*, 2002.

[7] C. Gehrmann, C. J. Mitchell, and K. Nyberg, "Manual authentication for wireless devices," *RSA Cryptobytes*, vol. 7, no. 1, pp. 29–37, 2004.

[8] S. Vaudenay, "Secure communications over insecure channels based on short authenticated strings," in *Advances in cryptology–CRYPTO 2005*. Springer, 2005, pp. 309–326.

[9] F.-L. Wong and F. Stajano, "Multi-channel protocols," in *Security Protocols*. Springer, 2007, pp. 112–127.