Analysis of a Mobile Communication Security Protocol

Tom Coffey, Reiner Dojen
Data Communication Security Laboratory
Department of Electronic and Computer Engineering
University of Limerick, Ireland

tom.coffey@ul.ie, reiner.dojen@ul.ie

Abstract: Cryptographic protocols are used to achieve secure communication over insecure networks. Weaknesses in such protocols are hard to identify, as they can be the result of subtle design flaws. Formal verification techniques provide rigid and thorough means to evaluate security protocols. This paper demonstrates the process of formal verification by applying a logic to a security protocol intended for use in mobile communications. As a result of the verification, 8 failed protocol goals are identified. Further, a new attack on the protocol is outlined. The presence of weaknesses in published protocols highlights the importance of formal verification to prevent insecure protocols reaching the public domain.

Keywords: Formal Verification, Security Protocols, Modal Logics, BCY Protocol

1. Introduction

Cryptographic protocols are used to achieve secure communications over insecure networks. The design of secure protocols is a highly complex and error-prone process. This is particularly evident from the surprisingly large number of published protocols which have later been found to contain various flaws, in many cases several years after the original publication [1], [2], [3].

Traditionally, cryptographic protocols have been designed and verified using informal and intuitive techniques. However, the absence of formal verification of these protocols can lead to flaws and weaknesses remaining undetected. Formal verification techniques provide a rigid and thorough means of evaluating the correctness of a cryptographic protocol so that even subtle defects can be discovered. Hence, they facilitate the design and testing of general-purpose cryptographic protocols [4].

This paper highlights the importance of using formal verification techniques prior to publication of security protocols. As a case study, the BCY protocol is verified using the GNY logic. This verification identifies 8 failed protocol goals. A new attack on the protocol is outlined.

The remainder of this paper is structured as follows: Section 2 summarises formal verification using modal logics. Section 3 presents a detailed verification of the BCY protocol and outlines a new attack. Section 4 concludes this paper.

2. Formal Verification

Formal verification of cryptographic protocols is essential as it supports the detection of defects in the protocol design. Even experienced security protocol designer find it

difficult to establish the correctness of security protocols, as protocol failure can result from very subtle defects.

2.1. Logic-based Verification

Logical techniques are bases on modal logics of belief and/or knowledge. Such logics involve a process of deductive reasoning. An attempt is made to derive the protocol goals by applying a set of axioms and inference rules to the assumptions and message exchanges of the protocol. Formal logics [1], [2], [3], [5], [6] can be used to generate comparably short and simple proofs. Logics have been used to identify a number of flaws in protocols previously considered to be secure. Logic-based formal verification involves the following steps:

1. Formalisation of the protocol messages
2. Specification of the initial assumptions
3. Specification of the protocol goals
4. Application of the logical postulates

A successfully verified protocol can be considered secure within the limitations of the logic. On the other hand, the results of a failed verification assist in the identification of missing initial assumptions and design-flaws of the protocol. The importance of formal verification cannot be overstated, as it prevents the publication of faulty security protocols by aiding the early discovery of design flaws.

2.2. The GNY Logic

The logic of Gong, Needham and Yahalom [2], usually referred to as the GNY logic, is used to formally verify cryptographic protocols. Discussions of the virtues and limitations of the logic can be found in [7] and [8]. Table 1 shows the constructs of GNY that are used throughout this paper. For a more detailed and complete description we refer to [2].

(X,Y)	Concatenation of formulae	$\phi(X)$	Formula X is recognizable
$\{X\}K$ $\{X\}K^{-1}$	Symmetric encryption and decryption	$P \ni X$	P possesses or is capable of possessing formula X.
$\{X\}K+$	Public key encryption/decryption	$P \mid\sim X$	P conveyed X.
$\{X\}K-$	Private key encryption/decryption	$P \mid\equiv X$	P believes X, i.e. the principal P acts as if X is true.
#(X)	The formula X is fresh. X has not been before the current run of the protocol.	X~>C	Message X has the extension C. The precondition for X being conveyed is C
$P \triangleleft *(X)$	P is told formula X, not conveyed by P during the current protocol run	$P \triangleleft X$	P is told X. P has a received a message containing X and P can read and repeat X.
$P \mid\Rightarrow X$	P has jurisdiction over X. The principal P is an authority on X.	$P \overset{K}{\longleftrightarrow} Q$	K is a suitable secret for P and Q. It may be used as a key or as a proof of identity.

Table 1. Used GNY constructs

3. The BCY Protocol and its Verification

The notation shown in Table 2 is used to describe the BCY protocol [9].

U	An identifier of the user	Kx+	public key of X
V	An identifier of the service provider	Kx-	private key of X
S	An identifier of the certification authority	KK	a key-encrypting key
r_X	a random nonce generated by X	SK	a session key
TS	An expiration time		

Table 2: BCY Protocol Notation

This protocol is aimed at providing authentication and key agreement in low-power portable devices such as mobile phones. It demonstrates the computational feasibility of using public-key cryptography in mobile communications. The protocol combines a Diffie-Hellman key exchange [10] with the modular square root (MSR) encryption technique [11]. The MSR technique reduces the computational burden placed on the mobile device. Certificates, denoted by $(X_1,\ldots,X_n)Ks-$, contain the components X_1 to X_n as well as a hash of these components signed by a certification authority. As the service provider uses two public keys, one for Diffie-Hellman key agreement and one for MSR encryption, these will be termed Kvd+ and Kvm+ respectively. The steps of the protocol are shown in Figure 1.

BCY1: $V \rightarrow U$: {V, Kvd+,Kvm+}Ks-
U computes $Y = \{r_U\}Kvm+$, KK={Kvd+}Ku-, $SK = \{r_U\}KK$
BCY2: $U \rightarrow V$: Y, {{U,Ku+}Ks-}r_U
V computes $r_U = \{Y\}Kvm-$, KK={Ku+}Kvd-, $SK = \{r_U\}KK$
BCY3: $V \rightarrow U$: {dataV}SK
BCY4: $U \rightarrow V$: {dataU}SK

Figure 1: The Original BCY Protocol

3.1. Verification of the BCY Protocol

A formalised version of the protocol is shown in Figure 2. The asterisks denote the ability of each principal to recognise that it did not send the received message at an earlier stage in the protocol. The message extensions in the first and second messages indicates that if S signs a certificate, then it believes that the public key contained in that certificate belongs to the principal, whose name is included in the certificate.

BCY1': $U \lhd *\{V,Kvd+,Kvm+\}Ks- \sim> S \mid\equiv \xrightarrow{Kvm+} V$, $S \mid\equiv \xrightarrow{Kvd+} V$
BCY2': $V \lhd *\{*r_U\}Kvm+$, $*\{*\{U,Ku+\}Ks- \sim> S \mid\equiv \xrightarrow{Ku+} U\ \}r_U$
BCY3': $U \lhd *\{dataV\}SK$
BCY4': $V \lhd *\{dataU\}SK$

Figure 2: The Formalised BCY Protocol

The initial assumptions for the BCY protocol are as follows:

U ∋ Ku-;	U ∋{U,Ku+}Ks-;	U ∋ dataU;	
V ∋ Kvd-;	V ∋ Kvm-;	V ∋ {V,Kvd+,Kvm+}Ks-;	V ∋ dataV;
U ∋ Ks+;	V ∋ Ks+;	U ∋ r_U;	U |≡ #r_U;
U |≡ ϕ(dataV) ;	V |≡ ϕ(dataU};	U |≡ ϕ(Kvd+,Kvm+);	V |≡ ϕ(Ku+);
U |≡ $\xrightarrow{Ks+}$ S;	U |≡ S ⇒ S |≡*;	U |≡(S ⇒ $\xrightarrow{Kvm+}$ V, S ⇒ $\xrightarrow{Kvd+}$ V);	
V |≡ $\xrightarrow{Ks+}$ S;	V |≡ S ⇒ S |≡*;	V |≡ S ⇒ $\xrightarrow{Ku+}$ U;	

The goals of the BCY protocol are as follows:

U |≡ $\xrightarrow{Kvd+}$ V;	U |≡ $\xrightarrow{Kvm+}$ V;	V |≡ $\xrightarrow{Ku+}$ U;
U ∋ SK;	U |≡ #SK;	U |≡ U $\xleftrightarrow{SK}$ V;
V ∋ SK;	V |≡ #SK;	V |≡ U $\xleftrightarrow{SK}$ V;
U |≡ V |~{dataV}SK;	U |≡ #{dataV}SK;	
V |≡ U |~{dataU}SK;	V |≡ #{dataU}SK;	

Application of the GNY postulates results in the following verification:

BCY1': U ◁ *{V,Kvd+,Kvm+}Ks- ~> S \|≡ $\xrightarrow{Kvm+}$ V, S \|≡ $\xrightarrow{Kvd+}$ V

Applying T1 to BCY1' yields U ◁ {V,Kvd+,Kvm+}Ks-. U is told V's certificate without the not-originated-here asterisk. Appling T6 yields U ◁ (V,Kvd+,Kvm+). U is told the contents of V's certificate. Appling T2 yields U◁Kvd+ and U◁Kvm+. U is told both of V's public keys. Applying P1 yields U ∋ Kvd+ and Kvm+. The user possesses the public keys of the service provider. Applying P8, we obtain U∋KK = {Kvd+}Ku-. U possesses the key–encrypting key. Since U possesses r_U, by applying P6, we obtain U ∋ SK = {r_U}KK. U possesses the session key. Since U believes that r_U is fresh, applying F1 gives U \|≡ #SK. U believes the session key is fresh. Since U recognises (Kvd+,Kvm+), by R1 U \|≡ ϕ(V,Kvd+,Kvm+). U recognises the contents of V's certificate. The prerequisites of I4 are satisfied. Thus, applying I4, we obtain U\|≡S\|~(V,Kvd+,Kvm+)Ks- and U\|≡S\|~(V,Kvd+,Kvm+). U believes that S conveyed V's certificate and the contents of the certificate. However, U cannot believe that the certificate is a valid current certificate of principal V. The preconditions of J2 are not achieved, as the certificate contains no expiration time. An intruder could use an old compromised certificate belonging to V in order to masquerade as the service provider. Thus, U cannot believe that Kvd+ and Kvm+ are V's current public keys. As Kvd+ is used to generate the session key, U cannot derive the belief that the session key, SK is a shared secret with V.

BCY2': V ◁ *{*r_U}Kvm+, *{*{U,Ku+}Ks- ~> S \|≡ $\xrightarrow{Ku+}$ U }r_U

Applying T1, T4 and P1 yields V ∋ r_U. The service provider possesses the random number generated by the user. Applying T1 and T3, we obtain V ◁ {U,Ku+}Ks-. V is told U's certificate. The proof now continues along similar lines to BCY1'. Applying T6, T2 and P1, we obtain V ∋ Ku+. V possesses U's public key. Applying P8, we obtain V∋KK = {Ku+}Kvd-. V possesses the key–encrypting key. Since V possesses r_U and KK by P6 we obtain V∋SK. V possesses the session key. However, V cannot establish that the session key is fresh. V recognises Ku+, by R1

V|≡ϕ(U,Ku+). V recognises the contents of U's certificate. The prerequisites of I4 are satisfied. Thus, applying I4, we obtain V|≡S|~(U,Ku+)Ks- and V|≡S|~ (U, Ku+). V believes that S conveyed U's certificate and the contents of the certificate. However, V cannot believe that the received certificate is a valid current certificate of principal U (for the same reasons that U cannot believe in the validity of V's certificate in BCY1'). Thus, V cannot believe that Ku+ is the public key of U or that the derived session key, SK, is a shared secret with U.

BCY3' and BCY4': U ◁ *{dataV}SK
V ◁ *{dataU}SK

Application of F7 yields U |≡ #{dataV}SK. U believes BCY3' to be fresh. In contrast, V cannot establish that BCY4' is fresh because he does not believe that the session key is fresh. Further, the final message exchange fails to provide authentication. The prerequisites for I1 are not satisfied as: (1) neither party believes that the session key is a shared secret and (2) V does not believe that the session key is fresh. Thus, neither principal believes that the other conveyed its identification data.

3.2. Weaknesses Identified by the Verification

The above verification of the BCY protocol identifies the following failed goals:

1. U cannot derive that V's public keys are valid
2. V cannot derive that U's public key is valid
3. U cannot derive that the session key is a suitable shared secret with V
4. V cannot derive that the session key is fresh
5. V cannot derive that the session key is a suitable shared secret with U
6. U cannot derive that V conveyed dataV.
7. V cannot derive that U conveyed dataU
8. V cannot derive that BCY4' is fresh.

Due to these failures, the BCY protocol provides neither authentication nor key agreement and hence, must be considered a failed protocol. Further, these failures allow an attacker to impersonate a legitimate user. As a precondition, the attacker must be able to obtain an old value of the user's nonce and the corresponding session key. It is worth noting that the availability of such information to an attacker was envisaged by the authors of the BCY protocol. With this information, the attacker, A, can replay the old nonce to the service provider V. This results in the inadvertent re-computation of an old session key, SK_O, by V. Subsequently, A can use the old session key for authentication to V, who would assume to communicate with the legitimate user U. This attack is possible, because V cannot determine freshness of the session key. Authentication and key agreement would only be achieved, if both parties were able to determine that the session key is fresh.

Some of these failures have been identified before and modified protocols have been proposed to resolve these failures [12],[13]. However, it has been shown that these modified protocols failed to rectify all failures and that the outlined attack can be applied to the new proposals as well [4].

The presence of weaknesses in the BCY protocol highlights the importance of formal verification. Even experienced security protocol designers have difficulties to ensure correctness of a protocol based on informal verification alone.

4. Concluding Remarks

In this paper, the formal verification of cryptographic security protocols using modal logics was discussed. The formal verification of a security protocol was demonstrated by applying a logic to a protocol intended for use in mobile communications.

Logic based verification and the GNY logic were briefly introduced. It followed a detailed verification of the BCY protocol. This verification compromises the translation of the protocol in the GNY logic, the identification of the initial assumptions and of the protocol goals and, finally, application of the GNY postulates. This verification identified 8 failed protocol goals, thus showing that the protocol fails to achieve authentication or key agreement. Further, a new attack was outlined, which allows an attacker to impersonate a legitimate user.

References

[1] M. Burrows, M. Abadi and R. Needham, "A logic of authentication", *ACM Operating System Review*, Vol. 23, No. 5, 1989, pp.1-13.

[2] L. Gong, R. Needham and R. Yahalom, "Reasoning about belief in cryptographic protocols", *1990 IEEE Computer Society Synopsis on Research in Security and Privacy*, 1990, pp.234-248.

[3] T. Coffey and P. Saidha, "A logic for verifying public key cryptographic protocols", *IEE Journal Proceedings-Computers and Digital Techniques*, Vol. 144, No. 1, 1997, pp.28-32.

[4] T. Coffey, R. Dojen and T. Flanagan, "Formal Verification: An Imperative Step in the Design of Security Protocols", To appear in *Journal on Computer Networks*, Elsevier Science, 2003

[5] V. Kessler and G. Wedel, "AUTLOG – An Advanced Logic of Authentication" *Proceedings of IEEE Computer Security Foundations Workshop IV*, 1994, pp. 90-99

[6] Y. Zhang and V. Varadharajan, "A Logic for Modelling the Dynamics of Beliefs in Cryptographic Protocols." *Australasian Computer Science Conference*, 2001.

[7] L. Gong, "Handling infeasible specifications of cryptographic protocols", *Proceedings of the 4th Computer Security Foundation Workshop*, 1991, pp.99-102.

[8] A. Mathuria, R. Safavi-Naini and P. Nickolas, "Some remarks on the logic of Gong, Needham and Yahalom", *Proceedings of the International Computer Symposium*, Vol. 1, 1994, pp.303-308.

[9] M.J. Beller, L.-F. Chang and Y. Yacobi, "Privacy and authentication on a portable communications system", *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 6, 1993, pp.821-829.

[10] W. Diffie and M.E. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory*, Vol. 22, 1976, pp. 644-654.

[11] M.O. Rabin, "Digitalized signatures and public-key functions as intractable as factorisation", *Technical Report MIT/LCS/TR-212*, MIT, 1979.

[12] U. Carlsen, "Optimal privacy and authentication on a portable communications system", *ACM Operating Systems Review*, Vol. 28, No. 3, July 1994, pp. 16-23.

[13] Y. Mu, and V. Varadharajan, "On the design of security protocols for mobile communications", *Information security and Privacy*, Lecture notes in Computer Science, Vol. 1172, 1996, pp. 134-145.