Securing Critical Unattended System with Identity Based Cryptography—A Case Study

Johannes Blömer, Peter Günther and Volker Krummel

Abstract. Unattended systems are key ingredients of various critical infrastructures like networks of self service terminals or automated teller machines. For cost and efficiency reasons they should mostly run autonomously. Unattended systems are attractive and lucrative targets for various kinds of attacks, including attacks on the integrity of their components and the communication between components. In this paper, we propose a general cryptographic framework to protect unattended systems. We also demonstrate that instantiating the framework with techniques from identity based cryptography is particularly well-suited to efficiently secure unattended systems.

1. Introduction

In this paper we present techniques from identity based cryptography (IBC) to secure unattended systems like automatic teller machines. An unattended system (USys) is a system that is designed to run autonomously without regular intervention by technical operators. Unattended systems form the core of many critical infrastructures. Examples of such systems can be found in nuclear power plants, industrial centers, self service terminals, and automated teller machines (ATMs). Usually, a USys consists of components that communicate via unprotected standard protocols like USB. Each component is an embedded device with limited computational, communication, and storage resources. Many USys are attractive and popular targets for attacks, e.g. a successful attack on ATMs may allow the attacker unauthorized access to cash. In other examples like control systems of industrial centers the damage caused by successful attacks will be even more severe. Hence the security of USys is an important task. Informally speaking, security of USys includes (at least) two aspects: 1) the integrity of all components of the system, 2) the authenticity, and in some cases the confidentiality, of the communication between the components of an unattended system.

Since the seminal paper by Boneh and Franklin [3], identity based cryptography (IBC) has emerged as one of the most powerful cryptographic technologies. Due to the intensive research over the past 10 years, basically, everything that can be achieved with tools from public-key cryptography, like RSA-based encryption schemes, RSA-based signature schemes or RSA-based key agreement protocols can also be achieved with the corresponding techniques from IBC (see [9] for an overview). However, IBC also provides tools that cannot be achieved with classical PKC, i.e. hierarchical encryption schemes. Moreover, whereas public-key cryptography relies on often complicated and expensive chains of certificates and public-key infrastructures, in IBC certified public keys can be replaced by secure identities. Hence, if secure identities are supplied by an application, independent of cryptographic purposes, IBC can be used to remove or at least simplify public-key infrastructures. Consequently, techniques from IBC have been proposed to secure diverse systems such as email [15], cloud computing [12, 8, 16], grid computing [13], and mobile ad-hoc networks (MANETs) [14, 11, 17].

M. Kosta and T. Sturm, eds. Proceedings of the MACIS 2013, Nanning, China, December 11-13, 2013.

Since USyss often provide secure identities, it seems natural to use identity-based cryptography to secure USys without relying on expensive public-key infrastructures. Hence, in this paper we design and describe IBC based security concepts and mechanisms for USys. We start with a precise description of the basic security requirements of USys. Then we show how cryptographic tools can be used to meet these requirements. Next, we instantiate the necessary tools with concrete schemes from IBC. Finally, we describe an IBC-based security system, that we have implemented for ATMs. The implementation includes efficient hardware implementations in secure environments of the main building block of most IBC primitives, i.e. bilinear pairings (see for example [7]). We conclude with a brief evaluation of our system and a summary of the lessons we have learned from implementing the system.

2. Unattended systems in the wild

USyss form the core of many critical infrastructures. Examples for such systems include control systems of nuclear power plants and industrial centers but also self service terminals and, in particular, automated teller machines (ATM). For instance networks of automated teller machines form the backbone of peoples cash supply network. To describe a security framework for unattended system we define an unattended system as an abstraction of the devices mentioned above. Hence, an *unattended system (USys)* is an IT-based system that runs autonomously without a regular intervention of a technical operator. In particular, any intervention by technical operators is business critical and has to be avoided.

A USys itself consists of components that communicate via standard protocols like USB. Each component can be regarded as an embedded (mechatronical) device with restricted computational power and storage. For example an ATM consists of components like the so called Electronic Pin Pad for entering the users PIN, a card reader for reading the customers banking card, a cash dispenser for handling bank notes and a system PC as a central control unit.

A large number of USys form the USys network, e.g., ATM network. A monitoring server can supervise the USys network remotely in order to get status information, do software updates and other administrative tasks. However, as explained above unattended systems are designed to run without permanent technical maintenance. Only exceptional circumstances may justify human interaction. This property has some non obvious effects on the security mechanisms for USys. I.e., one cannot ensure a direct interaction of a technical operator in case of a security breach.

Security model. Depending on its purpose, a USys is a popular target for attackers. In the example of an ATM an attacker wants to get unauthorized access to the cash. To reach this goal an attacker may attack the USys in different ways. For simplicity, in our threat model we only consider the following attack strategies:

Component Substition Attack. The attacker prepares a component as a substitute for a specific USys component. This substitute works as a regular component but also contains some malicious mechanisms. After exchanging a valid component by its manipulated substitute inside the USys, the attacker activates the malicious mechanisms in order to execute an unauthorized action, e.g. an unauthorized cash dispense.

Message Manipulation Attack. The attacker forces his way into the USys in order to get access to the communication lines of the components. Beside eavesdropping the communication the attacker is also able to manipulate and induce messages into the communication. The analysis of the underlying communication protocols let the attacker induce malicious messages that in turn execute unauthorized actions, e.g. unauthorized cash dispenses.

These attack scenarios together with the main features of a USys lead to the following security requirements.

Component Authenticity. The USys only consists of authentic components.

Data Origin Authenticity. The communication between components must be authenticated.

local verifiability. Detection and reaction on integrity breaches must rely only on the components inside the USys.

Avoid Single Points of Failure / Attack. Detection and reaction on integrity breaches must still be possible if individual components inside the USys fail.
Efficiency. Mechanisms function efficiently.

An unattended system is called secure or achieves *System Integrity* if it fulfills all these requirements.

2.1. Security framework for unattended systems

To design a framework that achieves System Integrity we propose two main steps:

1. Each component verifies the authenticity of every other component within the same USys (mutual integrity checks).
2. After successfully verifying the authenticity of another component a secure channel is established between these two components.

To realize this approach we propose to enhance the basically unprotected communication of a USys using the following cryptographic building blocks.

Identification Protocol. An identification protocol allows a prover to prove its identity to a verifier if the verifier has authenticated access to the public key of the prover.
Public Key Encryption. A public key encryption scheme allows confidential communication between a sender and a receiver if the sender has authenticated access to the public key of the receiver.
Digital Signature. A signature scheme allows authenticated communication between a prover and a verifier if the verifier has authenticated access to the public key of the prover.
Key Establishment. A key establishment protocol with key authentication that allows two parties to share an authenticated symmetric key if both parties are able to prove their identity.
Symmetric Encryption. A symmetric cipher allows confidential communication between two parties if they share an authenticated symmetric secret key.
Message Authentication Code. A message authentication code (MAC) allows authenticated communication between two parties if they share an authenticated symmetric secret key.

From the list of cryptographic building blocks, it becomes evident that authenticated access to public keys it crucial for setting up our system. As explained later in Section 3.1 this problem will be efficiently solved by our identity (ID) based approach.
Discussion of security framework. To design unattended system that achieve System Integrity we use these building blocks as follows.

1. Each component executes a cryptographic identification protocol with all other components. This protocol is secured by the use of the components keys. After the execution of the protocol each component knows the authenticity status of all other components.
2. If a component was proven to be authentic, a key establishment protocol is executed to establish an authentic and confidential communication channel for future communication. If a component failed to prove its authenticity, the other components refuse to set up a communication channel. Depending on the type of component, further reactions are possible like informing a monitoring server, refuse to work, turn off the unattended system etc., in order to reduce the damage caused by a malicious component.

Informally, the following arguments show that this approach leads to System Integrity for a USys.

1. The property of *component authenticity* is guaranteed by executing an identification protocol using a unique ID for every component. If required, the protocol is repeated within defined intervals to detect integrity breaches in a running system.
2. The property of *local verifiability* is implicitly fulfilled by the mutual authentication between all components.
3. By using a secure communication protocol based on symmetric encryption and a MAC for the communication between two components, the requirement of *data origin authenticity* is fulfilled.

4. The secure communication channels established after the component authenticity are purely based on symmetric cryptography. This enables a high throughput to meet the *efficiency* requirement of the USys.
5. By using the principle of pairwise mutual authentication we also avoid the introduction of a special security component as a *single point of failure*.

Before describing in detail, the specific IBC schemes we propose to use, we give a more high level description of (standardized) techniques from either public key cryptography (PKC) or IBC one can use to instantiate the cryptographic building blocks listed above.

To provide component authentication we propose to use a challenge-response protocol based on digital signatures. To define an authenticated key establishment protocol, we propose key transport mechanisms based on a public key encryption scheme and a signature scheme like in X.509. Specifically, we will use IBC to instantiate these schemes as explained later. As a second option we propose to directly define an identity based key establishment protocol with key authentication like in [5]. In general, the second option will result in more efficient schemes. Sometimes, as in our ATM scenario, the choice may be determined by the application. Here, standards like DIN66291 [1] have to be fulfilled and leave only few or just one option.

Secure communication is established in two steps. First, every pair of components that needs to communicate will execute the key establishment protocol to share the necessary private session keys. In the second step, one key pair will be used in the symmetric block cipher to establish an encrypted channel between the two components. Another key pair will be used in a MAC to establish an authenticated communication between the components. As an efficient alternative a special block cipher mode of operation that achieves both confidentiality and data authenticity can be used. One such mode of operation is the so called EAX mode [2].

2.2. Definition of IDs

To use techniques from IBC, it is important that every component is personalized with a secure unique identity. Among other things, this ID is required for addressing other components in the system or for backtracking components.

We model IDs such that they reflect the components positions within the hierarchy of the system. Take our ATM scenario as an example. The ID of a cash dispenser might consist of three components, a substring identifying the owner of the ATM, a substring identifying the ATM that hosts the dispenser, and a substring that describes the functionality of the component within the ATM: `CustomerXY/ATM-182641/Dispenser`. Knowing its own ID, every component can easily deduce the ID of other entities. If, for example, the cash dispenser has to talk to the electronic pin pad (EPP) of the same ATM it simply replaces the last substring to obtain `CustomerXY/ATM-182641/EPP`. Of course other approaches to define IDs like serial numbers or network IPs are possible.

3. Implementing security with identity based cryptography

In this section, we present our approach to ensure the system integrity of USyss as described in Section 2. Our approach is based on the techniques of IBC such as identity based signature (IBS) and identity based encryption (IBE). Therefore we start by giving a short introduction into IBC.

3.1. Background on identity based cryptography

IBC can be seen as a special version of PKC. In IBC and in contrast to classical PKC, public keys can be arbitrary strings. Typically, strings that uniquely identify the owner of the key are used, hence the name IBC. Since the public key is defined by the ID of its owner, it does not need to be published. This allows to dramatically reduce the complexity of the public key infrastructure (PKI) compared to classical approaches. Consequently, IBC is interesting for large scale systems with prescribed identities. Remark that in USyss all components have unique IDs as explained in Section 2.2.

As an example take IBE. To encrypt a message in an IBE system, the encryption algorithm takes as input the message, the ID of the recipient, and some globally defined public parameters. It

outputs a ciphertext. The ciphertext, together with the private key of the recipient is the input of the decryption algorithm that outputs the original message.

It is inevitable that the private key of every ID is computed based on some system-wide master secret key. Hence there is the need for a special party in the system that supervises this master key and computes the individual private keys. Since this distinguished party, often called private key generator (PKG), knows all private keys it is an attractive target for attacks.

IBC was proposed first in 1984 by A. Shamir. He also described identity based signatures. But it was not until 2001 that Boneh and Franklin presented the first fully functional identity based encryption scheme. The Boneh Franklin scheme as well as most state-of-the-art techniques in IBC use techniques from elliptic curve cryptography. However, in addition to the standard techniques from ECC IBC techniques rely on bilinear pairings.

3.2. A concrete identity based signature scheme

Since an identity based signature (IBS) scheme is one of our main building blocks, we briefly recall the definition of a concrete scheme [4] that we will use later. An IBS scheme consists of four algorithms *Setup*, *Extract*, *Sign*, and *Verify*. The algorithm *Setup* is used to setup all global system parameters and to generate the master secret key. On input an ID and the master secret key, the algorithm *Extract* outputs the secret key for ID. On input a message and a secret key for an ID, the algorithm *Sign* outputs a signature for the corresponding message and ID. To be correct, on input this message, this signature, and this ID, the algorithm *Verify* has to output `true`. For the scheme to be secure, no attacker should be able to generate a message and a signature for an uncorrupted ID that will cause *Verify* to output `true`. Here is the formal definition of the scheme:

Definition 3.1. The Cha-Cheon identity based signature scheme [4].

1. **Setup**(1^n): Choose cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of size at least 2^n and an efficiently computable, non-degenerate, bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Select a generator $g_2 \in \mathbb{G}_2$. Choose a random $s \in \mathbb{Z}_l$ as master secret key and $g_{pub} = g_2^s$. Define cryptographic hash functions $H_1 : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{Z}_l$ and $H_2 : \{0,1\}^* \to \mathbb{G}_1$. Define the public parameters (g_2, g_{pub}, H_1, H_2).
2. **Extract**(ID, H_2, s): Output $d_{ID} = H_2(ID)^s$ as private key for ID.
3. **Sign**$(m, H_1, H_2, d_{ID}, ID)$: Choose $r \in \mathbb{Z}_l$ uniformely at random. Compute $u = H_2(ID)^r$ and $v = d_{ID}^{r+H_1(m,u)}$. Output $\sigma = (u, v)$ as signature for message m and secret key d_{ID}.
4. **Verify**$(\sigma, ID, g_{pub}, g_2, H_1, H_2, m)$: Parse $\sigma = (u, v)$. Output true if and only if
$$e\left(uH_2(ID)^{H_1(m,u)}, g_{pub}\right) = e(v, g_2).$$

Bilinearity implies the correctness of the scheme:

$$\begin{aligned} e\left(uH_2(ID)^{H_1(m,u)}, g_{pub}\right) &= e\left(H_2(ID)^{rs} H_2(ID)^{H_1(m,u)s}, g_2\right) \\ &= e\left(H_2(ID)^{s(r+H_1(m,u))}, g_2\right) = e\left(d_{ID}^{r+H_1(m,u)}, g_2\right) = e(v, g_2). \end{aligned}$$

We see that the bilinearity of the pairing allows the verification of signatures using only public known identities by moving the master secret s from the public parameter g_2^s in $\mathbb{G}_2$ to the identity $H_2(ID)$ in $\mathbb{G}_1$.

The scheme is secure against *existential forgery on adaptively chosen message and ID attacks* in the *random oracle model* if the computational Diffie-Hellman problem is hard in the groups selected in the algorithm *setup* [4].

To instantiate the hash functions H_1 and H_2 in practice we propose the usage of SHA-2 as building block. To implement H_2 we need to hash into $\mathbb{G}_1$. How this could be done is explained in [3].

3.3. Encryption and message authentication schemes

We selected the signature scheme from [4] because it shares a lot of its functionality with the IBE scheme from [3]. Hence, we use the IBE scheme from [3] for asymmetric encryption as required in our key establishment protocol. This reduces the complexity of the cryptographic software components.

For the symmetric mechanisms we focused on standardized algorithms like AES [6] in EAX mode [2]. AES can be efficiently implemented for all relevant hardware platforms. The EAX mode uses only the AES encryption algorithm to achieve confidentiality and authentication.

3.4. Personalization

With its entry into the system, each component needs to be personalized with its unique ID. This is typically done at the production site during system integration. As described above, each component requires the private key belonging to its ID. For this step the master secret key and hence the PKG is required. Therefore, as for classical PKC, part of the personalization has to be done in a secured environment. A second personalization with a public key or with additional certificates is only necessary for classical PKC but not necessary in the IBC setting.

3.5. Maintenance and replacement of components

By Definition 2 a subsystem where a component is replaced should be considered as corrupted. This implies that if a component in the system has to be removed because of a technical failure, it has to be replaced by a component that is personalized with the same ID. In an identity based setting the private key of every component can simply be re-generated from the global master secret. Hence, there is no need for storing private keys at the production site.

4. Efficiency of the IBC based security framework

In this section we look at the efficiency of our proposed framework. Efficiency is the second important property of the security framework for unattended system. It covers computational and communication efficiency.

Computational Efficiency. As explained above, each component must execute the cryptographic protocols and the cryptographic primitives listed above. Hence, an efficient implementation of these primitives is crucial for the overall computational efficiency. Compared to classical asymmetric algorithms like RSA, the identity based cryptography approach achieves the same level of security with shorter keys. Depending on the parameters, IBC can achieve the same security level as RSA but with private key size comparable to elliptic curve cryptography (ECC) . This in turn yields efficient implementations if optimized for the underlying hardware. We implemented the IBE scheme from [3] as well as the IBS scheme from [4] on cryptographic hardware offering support for elliptic curve cryptography, but no specific support for IBC. Even with these restrictions we were able to implement schemes sufficiently efficient for our application of ATMs. For example, we are able to generate and verify signatures as in Definition 3.1 in a few hundred milliseconds on a state-of-the-art smartcard. Here the key length was chosen to provide security comparable to RSA signatures with a key length of 1024 bits. We expect further performance improvements if the underlying hardware fully supports the mathematical operations of IBC algorithms.

Communication Efficiency. Another important factor for efficiency is the communicational efficiency. To estimate communicational efficiency we compare the number and the size of message that have to be exchanged in both classical PKC and IBC approaches. Since in the IBC approach we simply adapt standard protocols from PKC, in the number of messages exchanged there is hardly any difference between the two approaches. The size of messages in the cryptographic protocols is determined by the size of the cryptographic key and by the size of additional information such as certificates that have to be included. As mentioned already above, keys in IBC tend to be shorter than keys in RSA-type PKC schemes, but larger than the keys in PKC schemes based on elliptic curve cryptography. However, in order to ensure entity or component authenticity in the classical PKC approach, both partners have to exchange certificate chains signed by trusted authorities. In the IBC there is no need to exchange certificates because the binding of a key to an entity or component is implicitly given by the secure identities. Since certificates in PKC are relatively large, messages in protocols based on IBC are significantly smaller than in protocols based on standard PKC protocols. Hence, the IBC approach increases communication efficiency when compared to the classical PKC approach.

5. Lessons learned & future work

Most of the work presented in this paper was done in the research project "SIS" funded by the German Ministry for research and Education[1]. In this project we learned that the effort to introduce identity based cryptography is very high in the beginning. This is mostly due to the complexity of the cryptographic algorithms itself. We had to implement all the algorithms from scratch. Here the challenge was to get efficient implementations that meet the mandatory time constraints of our application, i.e. ATMs.

However, after mastering this step, in several aspects IBC proved to be an improvement compared to classical PKC. The ratio of efficiency and security is far better than for the classical PKC. The security requirements for the supervising backend systems are lower. For example, there is no need to securely store a huge database of public keys on a key server. This reduces the complexity of the backend and in turn increases security and efficiency. Last but not least the security processes for maintaining the whole USys network are simplified. This also reduces error rates and improves the security and robustness.

One question that we still have to consider in our future work is the security and reliability of the private key generator (PKG) in IBC. As mentioned in Section 3, in IBC the PKG generates private keys from a master secret key and identities. Hence, the PKG as owner of the master secret key requires ultimate trust. Furthermore, it must be implemented securely and reliably. To achieve trust, security, and reliability in a PKG we will look at distributed realizations of PKGs as proposed for example in [10].

References

[1] DIN V66291-4 Chipkarte mit digitaler Signaturanwendung/Funktion nach SigV und SigG, 2002. A German standard for digital signatures on smartcards.

[2] Mihir Bellare, Phillip Rogaway, and David Wagner. The EAX Mode of Operation. In Bimal K. Roy and Willi Meier, editors, *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer, 2004.

[3] Dan Boneh and Matthew Franklin. Identity-based encryption from the weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[4] Jae Choon Cha and Jung Hee Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30. Springer, 2003.

[5] Liqun Chen, Zhaohui Cheng, and Nigel P. Smart. Identity-based key agreement protocols from pairings. *Int. J. Inf. Sec.*, 6(4):213–241, 2007.

[6] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002.

[7] Steven D Galbraith, Kenneth G Paterson, and Nigel P Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[8] Dijiang Huang, Zhibin Zhou, Le Xu, Tianyi Xing, and Yunji Zhong. Secure data processing framework for mobile cloud computing. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 614–618. IEEE, 2011.

[9] Marc Joye and Gregory Neven. *Identity-based cryptography*, volume 2. IOS Press, 2009.

[10] Aniket Kate and Ian Goldberg. Distributed private-key generators for identity-based cryptography. In *Security and Cryptography for Networks*, pages 436–453. Springer, 2010.

[11] Aram Khalili, Jonathan Katz, and William A Arbaugh. Toward secure key distribution in truly ad-hoc networks. In *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on*, pages 342–346. IEEE, 2003.

[12] Hongwei Li, Yuanshun Dai, and Bo Yang. Identity-based cryptography for cloud security. *IACR Cryptology ePrint Archive*, 2011:169, 2011.

[1] Grant number 01IS10030

[13] Hoon Wei Lim and Kenneth G Paterson. Identity-based cryptography for grid security. *International Journal of Information Security*, 10(1):15–32, 2011.

[14] Nitesh Saxena, Gene Tsudik, and Jeong Hyun Yi. Identity-based access control for ad hoc groups. In *Information Security and Cryptology–ICISC 2004*, pages 362–379. Springer, 2005.

[15] Diana K Smetters and Glenn Durfee. Domain-based administration of identity-based cryptosystems for secure email and ipsec. In *Proceedings of 12th Usenix Security Symposium*, volume 6, pages 6–5, 2003.

[16] Guojun Wang, Qin Liu, and Jie Wu. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 735–737. ACM, 2010.

[17] Shushan Zhao, Akshai Aggarwal, Richard Frost, and Xiaole Bai. A survey of applications of identity-based cryptography in mobile ad-hoc networks. *Communications Surveys & Tutorials, IEEE*, 14(2):380–400, 2012.

Johannes Blömer
University of Paderborn
e-mail: bloemer@upb.de

Peter Günther
University of Paderborn
e-mail: peter.guenther@upb.de

Volker Krummel
Wincor Nixdorf International
e-mail: volker.krummel@wincor-nixdorf.com