Minimalist Cryptography

M.Reza Sohizadeh A.

University Of Bergen

Norsk kryptoseminar 2011

Outline

1 Introduction

2 Minimal Cryptography

- Minimal Primitives
- Minimal Protocols

3 Conclusion

Wikipedia

- Minimalism describes movements in various forms of art and design, where the work is stripped down to its most fundamental features.
- The term minimalism is also used to describe a trend in design and architecture where in the subject is reduced to its necessary elements.

Minimalism

(a)

(b)

Figure: Minimalism

Minimal Cryptography vs Lightweight Cryptography

- Minimal Cryptography can be used in lightweight cryptography.

Minimal Cryptography vs Lightweight Cryptography

- Minimal Cryptography can be used in lightweight cryptography.
- But all lightweight schemes are not necessarily minimal.

Minimal Cryptography

- Minimal Primitives (Block cipher,...)
- Minimal Protocols (Authentication,...)

Minimal primitives

- Many papers were published on the minimal cryptographic assumptions which are necessary and sufficient in order to construct various types of secure primitives.

Minimal primitives

- Many papers were published on the minimal cryptographic assumptions which are necessary and sufficient in order to construct various types of secure primitives.
- We have to consider minimal schemes (which are local minima that become insecure when we eliminate any one of their elements) rather than minimum schemes (which are global minima among all the possible constructions).

Minimal Block Cipher

- What is the simplest possible construction of a block cipher which has a formal proof of security?

Minimal Block Cipher

- What is the simplest possible construction of a block cipher which has a formal proof of security?
- This problem was first addressed by Even and Mansour in 1991.

The Even-Mansour scheme

- They were motivated by the DESX construction, in which to protect DES against exhaustive search attacks we can XOR two independent pre-whitening and post-whitening keys to the plaintext and ciphertext (respectively).

The Even-Mansour scheme

- They were motivated by the DESX construction, in which to protect DES against exhaustive search attacks we can XOR two independent pre-whitening and post-whitening keys to the plaintext and ciphertext (respectively).
- The resultant scheme increased the key size from 56 to 184 bits without changing the definition of DES and with almost no additional complexity.

The Even-Mansour scheme

- The Even-Mansour scheme used such whitening keys but eliminated the keyed block cipher in the middle, replacing it with a fixed random permutation that everyone can share.

The Even-Mansour scheme

- The Even-Mansour scheme used such whitening keys but eliminated the keyed block cipher in the middle, replacing it with a fixed random permutation that everyone can share.

$$C = EM^{\mathcal{F}}_{K_1,K_2}(P) = \mathcal{F}(P \oplus K_1) \oplus K_2 \qquad (1)$$

Security Model

It is assumed that the adversary is allowed to perform two types of queries:

- Queries to a full encryption/decryption oracle, called an E-oracle, that computes either

$$E(P) = EM^{\mathcal{F}}_{K_1,K_2}(P) \tag{2}$$

or

$$D(C) = (EM^{\mathcal{F}}_{K_1,K_2})^{-1}(C). \tag{3}$$

Security Model

It is assumed that the adversary is allowed to perform two types of queries:

- Queries to a full encryption/decryption oracle, called an E-oracle, that computes either

$$E(P) = EM^{\mathcal{F}}_{K_1,K_2}(P) \tag{2}$$

or

$$D(C) = (EM^{\mathcal{F}}_{K_1,K_2})^{-1}(C). \tag{3}$$

- Queries to an F-oracle, that computes either $\mathcal{F}(x)$ or $\mathcal{F}^{-1}(y)$.

Security of the scheme

- The designers of EM considered two types of attacks.
  - Existential forgery attack, the adversary tries to find a new pair (P, C) such that $E(P) = C$.

Security of the scheme

- The designers of EM considered two types of attacks.
  - Existential forgery attack, the adversary tries to find a new pair (P, C) such that $E(P) = C$.
  - The adversary tries to decrypt a message C, i.e., to find P for which $E(P) = C$.

Security of the scheme

- The designers of EM considered two types of attacks.
  - Existential forgery attack, the adversary tries to find a new pair (P, C) such that $E(P) = C$.
  - The adversary tries to decrypt a message C, i.e., to find P for which $E(P) = C$.
- The data complexity of an attack on the scheme is determined by the number D of queries to the E-oracle.

Security of the scheme

- The designers of EM considered two types of attacks.
  - Existential forgery attack, the adversary tries to find a new pair (P, C) such that $E(P) = C$.
  - The adversary tries to decrypt a message C, i.e., to find P for which $E(P) = C$.
- The data complexity of an attack on the scheme is determined by the number D of queries to the E-oracle.
- The time complexity of the attack is lower bounded by the number$T$ of queries to the F-oracle.

Security of the scheme

- The main rigorously proven result was an upper bound of $O(DT/2^n)$ on the success probability of any cryptanalytic attack (of either type) on EM that uses at most D queries to the E-oracle and T queries to the F-oracle.

Security of the scheme

- The first proposed attack on the Even-Mansour scheme was published by Joan Daemen at Asiacrypt 1991 . Daemen used the framework of differential cryptanalysis.

Security of the scheme

- The first proposed attack on the Even-Mansour scheme was published by Joan Daemen at Asiacrypt 1991 . Daemen used the framework of differential cryptanalysis.
- Later by a new attack called slide with a twist which was developed by Alex Biryukov and David Wagner, and presented at Eurocrypt 2000. By taking two Even-Mansour encryptions, sliding one of them and reversing the other, they showed how to attack the scheme with known instead of chosen plaintexts.

Security of the scheme

- The first proposed attack on the Even-Mansour scheme was published by Joan Daemen at Asiacrypt 1991 . Daemen used the framework of differential cryptanalysis.
- Later by a new attack called slide with a twist which was developed by Alex Biryukov and David Wagner, and presented at Eurocrypt 2000. By taking two Even-Mansour encryptions, sliding one of them and reversing the other, they showed how to attack the scheme with known instead of chosen plaintexts.
- These two cryptanalytic attacks were thus complementary.

Security of the scheme

- In 2011 Dunkelman and Shamir presented the new Slidex attack and use it to obtain a tight bound on the security of the Even-Mansour scheme.

Security of the scheme

- In 2011 Dunkelman and Shamir presented the new Slidex attack and use it to obtain a tight bound on the security of the Even-Mansour scheme.
- This allows to mount the following attack, for any $d \leq n$:
  - Query the E-oracle at $2^{(d+1)/2}$ arbitrary values.
  - Choose 2^{n-d} arbitrary values $\Delta_1, \Delta_2, ...$ For each Δ_l, query the F- oracle at the values $\{P_i \oplus \Delta_l\},\ i = 1, 2, ..., 2^{(d+1)/2}$.

Security of the scheme

- In 2011 Dunkelman and Shamir presented the new Slidex attack and use it to obtain a tight bound on the security of the Even-Mansour scheme.
- This allows to mount the following attack, for any $d \leq n$:
  - Query the E-oracle at $2^{(d+1)/2}$ arbitrary values.
  - Choose 2^{n-d} arbitrary values $\Delta_1, \Delta_2, ...$ For each Δ_l, query the F- oracle at the values $\{P_i \oplus \Delta_l\},\ i = 1, 2, ..., 2^{(d+1)/2}$.
- They found that if $K_1 = K_2$ the security bounds remains intact.

$$C = EM^{\mathcal{F}}_{K_1}(P) = \mathcal{F}(P \oplus K_1) \oplus K_1 \tag{4}$$

Security of the scheme

Known Plaintext Attacks				
Attack	Data	Time	Memory	Tradeoff
Guess and determine [8]	2	2^n	2	—
Slide with a twist [4]	$2^{n/2}$	$2^{n/2}$	$2^{n/2}$	—
Slidex (Sect. 3.2)	2^d	2^{n-d}	2^d	$DT = 2^n$
Chosen Plaintext Attacks				
Attack	Data	Time	Memory	Tradeoff
Differential [5]	2^d	2^{n-d}	2^d	$DT = 2^n$
Adaptive Chosen Plaintext Attacks				
Attack	Data	Time	Memory	Tradeoff
Slidex (Sect. 6)	2^d	2^{n-d}	1	$DT = 2^n$ $(D \geq 2^{n/2})$

Figure: Comparison of Results on the Even-Mansour scheme

Minimal Protocols

- In this paper, we explore a notion of minimalist cryptography suitable for RFID tags. We consider the type of security obtainable in RFID devices with a small amount of rewritable memory, but very limited computing capability.

Minimal Protocols

- In this paper, we explore a notion of minimalist cryptography suitable for RFID tags. We consider the type of security obtainable in RFID devices with a small amount of rewritable memory, but very limited computing capability.
- Ari Juels described a protocol that provably achieves the properties of authentication and privacy and involves no computationally intensive cryptographic operations, and relatively little storage.

Minimal Protocols

- In this paper, we explore a notion of minimalist cryptography suitable for RFID tags. We consider the type of security obtainable in RFID devices with a small amount of rewritable memory, but very limited computing capability.
- Ari Juels described a protocol that provably achieves the properties of authentication and privacy and involves no computationally intensive cryptographic operations, and relatively little storage.
- The main goal is to show that standard cryptographic functionality is not needed to achieve stronger security in RFID tags.

But Why?

- The cost of rudimentary RFID tags promises to drop to roughly \$0.05, while tags as small as 0.4mm $\times$ 0.4mm, and thin enough to be embedded in paper are already commercially available.

But Why?

- The cost of rudimentary RFID tags promises to drop to roughly \$0.05, while tags as small as 0.4mm $\times$ 0.4mm, and thin enough to be embedded in paper are already commercially available.
- One of the most advanced of the current generation of small, inexpensive RFID tags is the Atmel TK5552. This tag has 992 bits of storage and a data transmission rate of about 100kB per sec. However, it costs as much as \$1.00 per unit.

But Why?

- The cost of rudimentary RFID tags promises to drop to roughly \$0.05, while tags as small as 0.4mm $\times$ 0.4mm, and thin enough to be embedded in paper are already commercially available.
- One of the most advanced of the current generation of small, inexpensive RFID tags is the Atmel TK5552. This tag has 992 bits of storage and a data transmission rate of about 100kB per sec. However, it costs as much as \$1.00 per unit.
- Projections on the likely resources in several years of RFID tags with cost in the vicinity of \$0.05 include several hundred bits of memory and somewhere between 5,000 and 10,000 logical gates.

Security Model

- Like normal users, adversaries in an RFID-system are physically constrained: They must have physical proximity to RFID tags in order to read (and therefore attack) them.

Security Model

- Like normal users, adversaries in an RFID-system are physically constrained: They must have physical proximity to RFID tags in order to read (and therefore attack) them.
- Such adversaries are necessarily weaker than in a traditional cryptographic setting.

Adversary Model

- Limited successive tag queries.

Adversary Model

- Limited successive tag queries.
- Limited interleaving:
  - The stationary attacker is limited due to mobility of tags:
  - The mobile attacker is limited since this requires shuttling back and forth between tags and legitimate readers.

The Minimal Protocol

- Let k be some value stored in a tag, i.e., $k = \alpha_i \cup \beta_i \cup \gamma_i$ where $i = 1...s$.
- For every value k, we maintain in the tag a vector:

$$\Delta_k = \{\delta_k^{(1)}, \delta_k^{(2)}, .., \delta_k^{(m)}\} \tag{5}$$

where The pad $\delta_k^{(1)}$ is used to update the tag value k.

$$k = k \oplus \delta_k^{(1)} \tag{6}$$

- We use the notation update$(\Delta_k, \tilde{\Delta}_k)$ to denote the function that updates (Δ_k and "overlays"it with $\tilde{\Delta}_k$).

The minimal protocol

Tag		**Verifier**
$d \leftarrow (c \bmod k) + 1$ $c \leftarrow c + 1$		

$\alpha' \leftarrow \alpha_d$	$\xrightarrow{\alpha'}$	if α' is valid α_i for some tag T_x then
		$tag \leftarrow x$
		$\beta' \leftarrow \beta_i$
		$\gamma \leftarrow \gamma_i$
		mark α_i as invalid for T_x
		else
		output("reject") and abort

	$\xleftarrow{\beta'}$	
if $\beta' \neq \beta_d$ then output("reject") and abort		
$\gamma' \leftarrow \gamma_d$	$\xrightarrow{\gamma'}$	if $\gamma' \neq \gamma$ or $\gamma' = \perp$ then output("reject") and abort $\tilde{\Delta}_{ABC} \in_R \{\{0,1\}^l\}^{3km}$
	$\xleftarrow{\tilde{\Delta}_{ABC}}$	
		output(tag, "accept")
$\{\mathsf{update}(\Delta_\kappa, \tilde{\Delta}_\kappa)\}_{\kappa \in ABC}$ $\{\kappa \leftarrow \mathsf{pad}(\kappa, \Delta_\kappa)\}_{\kappa \in ABC}$		$\{\mathsf{update}(\Delta_\kappa, \tilde{\Delta}_\kappa)\}_{\kappa \in ABC}$ $\{\kappa \leftarrow \mathsf{pad}(\kappa, \Delta_\kappa)\}_{\kappa \in ABC}$

Figure: The minimal protocol

An Example

Reader $\mathcal{R}$ $(IDS_i, K1_i, K2_i), ID_j \in$ dB		**Tag** $\mathcal{T}_j$ $(IDS_{old}, K1_{old}, K2_{old})$, $(IDS_{next}, K1_{next}, K2_{next}), ID_j \in$ dB
	$\xrightarrow{hello}$	$IDS = IDS_{next}$, $K1 = K1_{next}$, $K2 = K2_{next}$. If *hello* received a second time,
	$\xleftarrow{IDS}$	then $IDS = IDS_{old}$, $K1 = K1_{old}$, $K2 = K2_{old}$.
If not $\exists IDS : (IDS_i, K1_i, K2_i) \in$ dB: resend *hello* to $\mathcal{T}_j$ and wait for IDS. Else: $IDS = IDS_i$, $K1 = K1_i$, $K2 = K2_i$. Randomly generate $n1$, $n2$. $A = IDS \oplus K1 \oplus n1$. $B = (IDS \vee K2) + n2$. $\hat{K1} = (K1 \oplus n2) << K1$. $\hat{K2} = (K2 \oplus n1) << K2$. $C = (K1 \oplus \hat{K2}) + (\hat{K1} \oplus K2)$.		
	$\xrightarrow{A\|\|B\|\|C}$	$n1 = A \oplus IDS \oplus K1$. $n2 = B - (IDS \vee K2)$. $\hat{K1} = (K1 \oplus n2) << K1$. $\hat{K2} = (K2 \oplus n1) << K2$. $\hat{C} = (K1 \oplus \hat{K2}) + (\hat{K1} \oplus K2)$. If $C = \hat{C}$:
	$\xleftarrow{D}$	$D = (\hat{K2} + ID_i) \oplus ((K1 \oplus K2) \vee \hat{K1})$. $IDS_{old} = IDS$, $IDS_{next} = (IDS + ID_i) \oplus (n2 \oplus \hat{K1})$. $K1_{old} = K1$, $K2_{old} = K2$. $K1_{next} = \hat{K1}$, $K2_{next} = \hat{K2}$.
$\hat{D} = (\hat{K2} + ID_i) \oplus ((K1 \oplus K2) \vee \hat{K1})$ If $D = \hat{D}$: $IDS_i = (IDS_i + ID_i) \oplus (n2 \oplus \hat{K1})$ $K1_i = \hat{K1}$, $K2_i = \hat{K2}$.		

Figure: The SASI protocol

Conclusion

- We explored two frameworks for minimal cryptography, one framework for block ciphers and one for ultra-lightweight mutual authentication protocol.

Conclusion

- We explored two frameworks for minimal cryptography, one framework for block ciphers and one for ultra-lightweight mutual authentication protocol.
- The schemes are both minimal and provably secure.

Conclusion

- We explored two frameworks for minimal cryptography, one framework for block ciphers and one for ultra-lightweight mutual authentication protocol.
- The schemes are both minimal and provably secure.
- There is still a lack of secure practical schemes in these frameworks.